



UNIVERSIDAD ESAN  
FACULTAD DE INGENIERÍA  
INGENIERÍA DE TECNOLOGÍAS DE INFORMACIÓN Y SISTEMAS

Predicción del éxito de financiamiento de un proyecto tecnológico en la plataforma de crowdfunding Kickstarter empleando técnicas de Aprendizaje Automático

Trabajo de investigación para el curso de Trabajo de Tesis II

Alonso Augusto Puente Ríos  
Asesor: Marks Arturo Calderón Niñuín

Lima, 18 de diciembre de 2019

Esta tesis denominada:

**PREDICCIÓN DEL ÉXITO DE FINANCIAMIENTO DE UN PROYECTO  
TECNOLÓGICO EN LA PLATAFORMA DE CROWDFUNDING KICKSTARTER  
EMPLEANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO**

ha sido aprobada.

---

(Jurado Presidente)

---

(Jurado)

---

(Jurado)

Universidad ESAN  
2019

PREDICCIÓN DEL ÉXITO DE FINANCIAMIENTO DE UN PROYECTO  
TECNOLÓGICO EN LA PLATAFORMA DE CROWDFUNDING KICKSTARTER  
EMPLEANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

## **AGRADECIMIENTO Y DEDICATORIA**

Durante la inducción en la empresa en la cual realicé mis segundas prácticas pre-profesionales, se realizaron varias actividades, entre ellas, una que me marcó positivamente. Esta consistía en comparar los tiempos de llegada de un punto a otro de una persona corriendo. Se caracterizó porque quien asumió el reto tuvo presente en su mente las personas y las razones por las cuales todos los días lucha y son su principal fuente de motivación.

Por ello, quiero dedicar este gran esfuerzo personal de trabajo de tesis a quienes siempre han estado a mi lado en los mejores y peores momentos, aquellos críticos en que definen el destino. Mi amada hermana Clarisabel, mis queridos padres Augusto e Isabel, mi familia en especial mis abuelos; y mis pocos, pero verdaderos y leales amigos de la universidad, colegio y trabajo. Todos ellos han sido y son cada uno, piedra fundamental en el desarrollo de mi ser como persona y profesional, así como también seres con los cuales siempre comparto gratos momentos. Su presencia en mi vida no ha sido una suerte más sino parte de mi destino.

Asimismo, luchar por mis sueños y mi país, y pensar cada día en solidificar su planificación me motivan emocionalmente hasta en aquellos momentos en que parece haber imposibles.

Quiero concluir esta sección, muy especial para mí, agradeciendo también a mi alma máter, la Universidad Esan, y al Programa Nacional de Becas (Pronabec) por hacer que estos 5 años entre el 2015 y 2019 sean mágicos y muy fructíferos. Tuve la oportunidad no solo de incrementar y potenciar mis conocimientos en distintas áreas académicas sino también de aprender de excelentes profesionales como mis profesores, conocer grandes amigos dentro y fuera de su campus (desde el primer ciclo como cachimbo hasta el último ciclo, en el CADE Universitario 2019, estudiantes de diferentes universidades y otras partes del Perú), ponerme a prueba en el exterior (en el II Congreso Internacional de Investigación en Colombia) y formar parte de la gran familia UE.

Por todos ellos, simplemente gracias.

<b>ÍNDICE</b>	
RESUMEN	11
INTRODUCCIÓN	12
1. CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	13
1.1. Descripción de la Realidad Problemática	13
1.2. Formulación del Problema	16
1.2.1. Problema General	16
1.2.2. Problemas Específicos	16
1.3. Objetivos de la Investigación	16
1.3.1. Objetivo General	16
1.3.2. Objetivos Específicos	16
1.4. Justificación de la Investigación	17
1.4.1. Teórica	17
1.4.2. Práctica	17
1.4.3. Metodológica	17
1.5. Delimitación del Estudio	18
1.5.1. Espacial	18
1.5.2. Temporal	18
1.5.3. Conceptual	18
1.6. Hipótesis	18
1.6.1. Hipótesis General	18
1.6.2. Hipótesis Específicas	18
1.6.3. Matriz de Consistencia	19
2. CAPÍTULO II: MARCO TEÓRICO	20
2.1. Antecedentes de la investigación	20
2.2. Bases Teóricas	35
2.2.1. Inteligencia Artificial	35
2.2.2. Aprendizaje Automático	37
2.2.3. Aprendizaje Profundo	41
2.2.4. Modelo Predictivo	42
2.2.5. Minería de Datos	43
2.2.6. Metodologías de Minería de Datos	44
2.2.7. Técnicas de Minería de Datos	48
2.3. Marco Conceptual	71
2.3.1. Crowdfunding	71
2.3.2. Kickstarter	71
2.3.3. Proyecto	72

2.3.4. Campaña	72
<b>3. CAPÍTULO III: METODOLOGÍA DE LA INVESTIGACIÓN</b>	<b>74</b>
3.1. Diseño de la Investigación	74
3.2. Población y muestra	77
3.3. Operacionalización de variables	79
3.4. Instrumentos de medida	81
3.5. Técnicas de recolección de datos	85
3.6. Técnicas para el procesamiento y análisis de la información	86
3.7. Cronograma de actividades y presupuesto	92
<b>4. CAPÍTULO IV: DESARROLLO DEL EXPERIMENTO</b>	<b>99</b>
4.1. Construcción de los conjuntos finales de datos	99
4.1.1. Metainformación	99
4.1.2. Contenido visual	104
4.1.3. Contenido textual	105
4.2. Análisis exploratorio de los datos	108
4.2.1. Metainformación	108
4.2.2. Contenido visual	113
4.2.3. Contenido textual	113
4.3. Pre-procesamiento de los conjuntos de datos	114
4.3.1. Metainformación	114
4.3.2. Contenido visual	115
4.3.3. Contenido textual	116
4.4. Creación de los modelos predictivos	120
4.4.1. Metainformación	120
4.4.2. Contenido visual	122
4.4.3. Contenido textual	126
<b>5. CAPÍTULO V: ANÁLISIS Y DISCUSIÓN DE RESULTADOS</b>	<b>128</b>
5.1. Metainformación	128
5.2. Contenido visual	132
5.3. Contenido textual	136
5.4. Demostración de modelos	140
<b>6. CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES</b>	<b>143</b>
6.1. Conclusiones	143
6.2. Recomendaciones	145
<b>BIBLIOGRAFÍA</b>	<b>147</b>
<b>ANEXOS</b>	<b>157</b>

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Resultados y ratios obtenidos en la encuesta por GEM y ESAN .....	13
<b>Figura 2.</b> Ratio de éxito de proyectos en Kickstarter (2017) .....	15
<b>Figura 3.</b> Marco de trabajo representado en 3 partes: metainformación (parte superior), contenido visual (medio) y contenido textual (parte inferior) .....	34
<b>Figura 4.</b> Ejemplo de algoritmo de regresión .....	38
<b>Figura 5.</b> Ejemplo de algoritmo de clasificación .....	38
<b>Figura 6.</b> Algoritmo de K Vecinos más cercanos con pesos ponderados .....	39
<b>Figura 7.</b> Funcionamiento del algoritmo de K medias .....	40
<b>Figura 8.</b> Componentes del Aprendizaje por Refuerzo .....	41
<b>Figura 9.</b> Diferencia entre Aprendizaje Automático y Aprendizaje Profundo .....	42
<b>Figura 10.</b> Diagrama de los seis pasos básicos .....	43
<b>Figura 11.</b> Fases de la metodología CRISP-DM .....	44
<b>Figura 12.</b> Fases de la metodología SEMMA .....	45
<b>Figura 13.</b> Fases de la metodología KDD .....	46
<b>Figura 14.</b> Modelo para representar una neurona propuesto por McCulloch y Pitts (1943) ..	49
<b>Figura 15.</b> Nodos con funciones de activación umbral en forma de puertas lógicas .....	50
<b>Figura 16.</b> Función de activación sigmoide .....	50
<b>Figura 17.</b> Ilustración del algoritmo gradiente descendiente .....	52
<b>Figura 18.</b> Actualización de pesos W con el algoritmo .....	53
<b>Figura 19.</b> Capa oculta simple MLP con propagación hacia atrás .....	54
<b>Figura 20.</b> Redes neuronales de ejemplo .....	54
<b>Figura 21.</b> Función de activación tangente hiperbólica .....	56
<b>Figura 22.</b> Función de activación puramente lineal .....	56
<b>Figura 23.</b> Función de activación Unidad Lineal Rectificada .....	57
<b>Figura 24.</b> Ejemplo de perceptrón simple .....	58
<b>Figura 25.</b> Ejemplo de perceptrón multicapa .....	58
<b>Figura 26.</b> Ejemplo de red neuronal convolucional .....	59
<b>Figura 27.</b> Modelo Necognitron de Fukushima (1980) .....	60
<b>Figura 28.</b> Modelo LeNet-5 de LeCun (1998) .....	60
<b>Figura 29.</b> Ejecución de la convolución en una entrada .....	61
<b>Figura 30.</b> Generación de una nueva imagen a partir de filtros .....	62
<b>Figura 31.</b> Secuencia de varias capas convolucionales .....	62
<b>Figura 32.</b> Extracción de características a partir de convoluciones .....	63
<b>Figura 33.</b> Ejemplo de matriz de imagen de entrada y un filtro .....	63
<b>Figura 34.</b> Dimensiones de una entrada y un filtro .....	64
<b>Figura 35.</b> Paso de 2 píxeles por parte de un filtro .....	65
<b>Figura 36.</b> Aplanado de matrices luego de agrupar la capa .....	66
<b>Figura 37.</b> Arquitectura completa de una CNN .....	66
<b>Figura 38.</b> Ejemplo de red neuronal recurrente .....	67
<b>Figura 39.</b> Hiperplano con dos clases separadas por una distancia $m$ .....	67
<b>Figura 40.</b> Ejemplo de caso linealmente separable .....	68
<b>Figura 41.</b> Ejemplo de caso no linealmente separable .....	69
<b>Figura 42.</b> Aplicación de un kernel para transformar el espacio de los datos .....	69
<b>Figura 43.</b> Ejemplo del algoritmo de árbol de decisión .....	70
<b>Figura 44.</b> Marco de trabajo del prototipo final .....	75
<b>Figura 45.</b> Arquitectura de un modelo de red VGG-19 .....	76
<b>Figura 46.</b> Arquitectura detallada de un modelo de red VGG-19 .....	77
<b>Figura 47.</b> Descripción de resultados de modelo descriptivo de ejemplo .....	83

<b>Figura 48.</b> Comparación de tres resultados de la curva AUC en el modelo.....	84
<b>Figura 49.</b> Cronograma de actividades del proyecto solución.....	93
<b>Figura 50.</b> Vista de la página de data disponible recolectada de Kickstarter.....	99
<b>Figura 51.</b> Tamaño y columnas del conjunto de datos, periodo de julio del 2019.....	100
<b>Figura 52.</b> Conjunto de datos filtrado por categoría, periodo de julio del 2019.....	100
<b>Figura 53.</b> Parte superior del flujo de trabajo de la data final.....	101
<b>Figura 54.</b> Parte inferior del flujo de trabajo de la data final.....	102
<b>Figura 55.</b> Visualización del archivo de metainformación subido a Kaggle.....	103
<b>Figura 56.</b> Proceso de descarga de imágenes.....	104
<b>Figura 57.</b> Visualización del archivo comprimido de imágenes en Kaggle.....	105
<b>Figura 58.</b> Función del algoritmo web scraping de la descripción de proyectos.....	106
<b>Figura 59.</b> Fraccionamiento de la data total en tres partes.....	106
<b>Figura 60.</b> Repetición del proceso para las fracciones restantes.....	107
<b>Figura 61.</b> Visualización del archivo de descripciones subido a Kaggle.....	107
<b>Figura 62.</b> Distribución total de proyectos tecnológicos por su estado.....	108
<b>Figura 63.</b> Evolución de cantidad de proyectos tecnológicos por año.....	108
<b>Figura 64.</b> Evolución de proyectos tecnológicos, por su estado y año.....	109
<b>Figura 65.</b> Matriz de correlaciones entre variables independientes.....	109
<b>Figura 66.</b> Matriz de correlaciones e histogramas de variables independientes.....	110
<b>Figura 67.</b> Caja de bigotes de la variable backers_count.....	111
<b>Figura 68.</b> Caja de bigotes de la variable goal.....	111
<b>Figura 69.</b> Caja de bigotes de la variable pledged.....	112
<b>Figura 70.</b> Caja de bigotes de la variable duration.....	112
<b>Figura 71.</b> Distribución de clases de dimensiones de imágenes de proyectos.....	113
<b>Figura 72.</b> Estadísticas de los datos normalizados.....	114
<b>Figura 73.</b> Conjunto de datos de metainformación de proyectos.....	116
<b>Figura 74.</b> Conjunto de datos de descripciones de proyectos.....	117
<b>Figura 75.</b> Conjunto X generado para el modelo de descripciones (izquierda).....	117
<b>Figura 76.</b> Conjunto Y generado para el modelo de descripciones (derecha).....	117
<b>Figura 77.</b> Flujograma de limpieza de conjunto de datos de descripciones.....	118
<b>Figura 78.</b> Proceso de limpieza de data de descripciones.....	119
<b>Figura 79.</b> Nube de palabras del contenido textual total.....	119
<b>Figura 80.</b> Ejemplo de parámetro $\gamma = 0.1$ .....	120
<b>Figura 81.</b> Ejemplo de parámetro $C = 0.1$ .....	120
<b>Figura 82.</b> Arquitectura del modelo de Red Multicapa para la metadata.....	121
<b>Figura 83.</b> Arquitectura del modelo final para las imágenes.....	122
<b>Figura 84.</b> Arquitectura del modelo VGG-19 con pesos pre-entrenados.....	124
<b>Figura 85.</b> Muestra del vocabulario de palabras de las descripciones.....	126
<b>Figura 86.</b> Ejemplo de funcionamiento del método BoW.....	126
<b>Figura 87.</b> Matriz de confusión del modelo SVM para la metadata.....	128
<b>Figura 88.</b> Puntaje AUC del modelo SVM para la metadata.....	129
<b>Figura 89.</b> Matriz de confusión del modelo MLP para la metadata.....	130
<b>Figura 90.</b> Puntaje AUC del modelo MLP para la metadata.....	131
<b>Figura 91.</b> Puntaje AUC del modelo con 350 épocas.....	131
<b>Figura 92.</b> Pérdida del modelo MLP con 350 épocas.....	132
<b>Figura 93.</b> Matriz de confusión del modelo VGG-19 para las imágenes.....	132
<b>Figura 94.</b> Puntaje del área AUC bajo la curva ROC del modelo de imágenes.....	133
<b>Figura 95.</b> Puntaje AUC del modelo con 250 épocas para las imágenes.....	134
<b>Figura 96.</b> Puntajes de pérdida del AUC con 250 épocas para las imágenes.....	134
<b>Figura 97.</b> Puntaje de exactitud del modelo con 150 épocas para las imágenes.....	135

<b>Figura 98.</b> Puntaje de pérdida de exactitud con 150 épocas para las imágenes.....	136
<b>Figura 99.</b> Matriz de confusión del modelo de SVM usando TF-IDF para las descripciones.....	136
<b>Figura 100.</b> Puntaje AUC del modelo SVM usando TF-IDF para las descripciones .....	137
<b>Figura 101.</b> Matriz de confusión del modelo de SVM usando BoW para las descripciones.....	138
<b>Figura 102.</b> Puntaje AUC del modelo SVM usando BoW para las descripciones .....	139
<b>Figura 103.</b> Imagen y metainformación de proyecto de muestra a predecir su estado.....	140
<b>Figura 104.</b> Parte de la descripción del proyecto de muestra a predecir su estado.....	140
<b>Figura 105.</b> Demo del modelo de metainformación con datos del proyecto de muestra.....	141
<b>Figura 106.</b> Demo del modelo de imágenes con la imagen re-dimensionada del proyecto de muestra.....	141
<b>Figura 107.</b> Demo del modelo de descripciones con la descripción del proyecto de muestra.....	142

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Matriz de Consistencia.....	19
<b>Tabla 2.</b> Cuadro comparativo entre características de las tres metodologías.....	47
<b>Tabla 3.</b> Distribución de los registros para el segundo experimento. ....	78
<b>Tabla 4.</b> Distribución de los registros para el tercer experimento. ....	78
<b>Tabla 5.</b> Descripción de las variables a usar para el conjunto de datos final.....	80
<b>Tabla 6.</b> Matriz de Confusión. ....	81
<b>Tabla 7.</b> Descripción de las variables del conjunto de datos recolectado. ....	91
<b>Tabla 8.</b> Presupuesto de los costos personales del autor.....	94
<b>Tabla 9.</b> Cuadro comparativo entre precios de servidores en la nube de Alibaba y Amazon Web Service (AWS). .....	95
<b>Tabla 10.</b> Descripción de características y precios de GPU dedicada en Paperspace. ....	96
<b>Tabla 11.</b> Cuadro comparativo entre instancias CPU, GPU y TPU en Paperspace.....	96
<b>Tabla 12.</b> Cuadro comparativo de diferentes servicios en la nube (Google Cloud, AWS, vast.ai, Paperspace). .....	97
<b>Tabla 13.</b> Comparación de precios de distintas instancias de Azure. ....	98
<b>Tabla 14.</b> Comparación de los resultados de todos los modelos.....	139
<b>Tabla 15.</b> Resultados de las demos de modelos entrenados con proyecto de muestra. ....	142

## ÍNDICE DE ECUACIONES

<b>Ecuación 1.</b> Cálculo de los pesos para el algoritmo K-NN mediante ponderación de sus distancias.....	39
<b>Ecuación 2.</b> Fórmula alternativa del algoritmo K-NN mediante sumatoria de pesos.....	39
<b>Ecuación 3.</b> Fórmula del algoritmo k-means. ....	40
<b>Ecuación 4.</b> Fórmula del cálculo del valor de un nodo i.....	49
<b>Ecuación 5.</b> Fórmula de una función de activación g para la salida del nodo. ....	49
<b>Ecuación 6.</b> Fórmula de la función de activación sigmoide. ....	50
<b>Ecuación 7.</b> Fórmula de función de coste de una regresión logística. ....	51
<b>Ecuación 8.</b> Actualización de pesos W mediante gradiente descendiente. ....	52
<b>Ecuación 9.</b> Fórmula del algoritmo de propagación hacia atrás. ....	53
<b>Ecuación 10.</b> Cálculo del error cometido en una red neuronal. ....	55
<b>Ecuación 11.</b> Actualización de pesos mediante propagación hacia atrás. ....	55
<b>Ecuación 12.</b> Cálculo de errores de nodos usando pesos actualizados. ....	55
<b>Ecuación 13.</b> Fórmula de la función de activación tangente hiperbólica. ....	56
<b>Ecuación 14.</b> Fórmula de la función de activación puramente lineal. ....	56
<b>Ecuación 15.</b> Fórmula de la función de activación ReLU. ....	57
<b>Ecuación 16.</b> Fórmula matemática de la convolución. ....	61
<b>Ecuación 17.</b> Cálculo del volumen del mapa de activación.....	63
<b>Ecuación 18.</b> Cálculo del tamaño de la imagen reducida. ....	64
<b>Ecuación 19.</b> Cálculo del tamaño de la imagen reducida con bordes llenos de ceros. ....	65
<b>Ecuación 20.</b> Ecuación del hiperplano para clasificar dos clases. ....	69
<b>Ecuación 21.</b> Fórmula para calcular la exactitud. ....	82
<b>Ecuación 22.</b> Fórmula para calcular la precisión. ....	83
<b>Ecuación 23.</b> Fórmula para calcular la sensibilidad. ....	85
<b>Ecuación 24.</b> Fórmula para calcular el puntaje F1.....	85
<b>Ecuación 25.</b> Fórmula del escalador Min-Max. ....	114
<b>Ecuación 26.</b> Fórmula del TF-IDF.....	127

## RESUMEN

Conocer el destino del financiamiento de proyectos siempre ha sido el principal deseo de todos los emprendedores que los promocionan en Internet, en especial, de la categoría de tecnología por ser los que presentan las ratios más bajas de éxito debido a sus altas metas que buscan alcanzar. El presente trabajo de investigación se basó en construir un modelo predictivo cuyo objetivo es la de estimar el éxito o fracaso de financiamiento de proyectos tecnológicos en la plataforma de crowdfunding Kickstarter durante la duración de su campaña a partir de su metainformación, imagen y/o descripción. Para ello, se crearon modelos de Aprendizaje Automático (SVM, MLP y CNN) para cada una de estas partes. Luego de analizar todos los modelos con las mismas métricas mencionadas en el décimo antecedente, se concluyó que solo los de metainformación tuvieron niveles excelentes de acuerdo a sus puntajes AUC (0.8377 para SVM y 0.7043 para MLP), mientras que los modelos de descripciones tuvieron un rendimiento regular (0.6746 para SVM con TF-IDF y 0.6709 para SVM con BoW) y finalmente el modelo de imágenes no logró clasificar las clases correctamente. Como trabajo a futuro, estos últimos modelos de descripción e imagen serán mejorados para construir, junto con los de metainformación, un modelo ensamblado basado en considerables variables del proyecto.

**Palabras clave:** *metainformación, descripción, imagen del proyecto, Máquina de Vectores de Soporte (SVM), Perceptrón Multicapa (MLP), Red Neuronal Convolutacional (CNN)*.

Knowing funding projects destiny has always been the main desire of all entrepreneurs who promote them on the Internet, especially in the technology category because they have the lowest success rates due to their high goals. The present research work was based on building a predictive model whose objective is to estimate the success or failure of technological projects funding in the Kickstarter crowdfunding platform during the duration of its campaign based on its metadata, image and/or description. For this, Machine Learning models (SVM, MLP and CNN) were created for each of these parts. After analyzing all the models with the same metrics mentioned in the tenth antecedent, it was concluded that only metadata models had excellent levels according to their AUC scores (0.8377 for SVM and 0.7043 for MLP), while the description models had a regular performance (0.6746 for SVM with TF-IDF and 0.6709 for SVM with BoW) and finally the image model failed to classify the classes correctly. As future work, these latest models of description and image will be improved to build, together with metadata ones, an assembled model base don considerable project variables.

**Keywords:** *project metadata, project description, project image, Support Vector Machine (SVM), Multilayer Perceptron (MLP), Convolutional Neural Network (CNN)*.

## INTRODUCCIÓN

Por muchos años, en especial en las dos últimas décadas, diversos proyectos emprendedores han sido lanzados en distintas plataformas web, buscando un objetivo compartido por todos: ser financiados en un determinado plazo para hacer realidad estas ideas. Entre fracasos y éxitos, han surgido nuevas tendencias, así como nuevos enfoques de estudios de estos casos para encontrar la clave que descifre las variables de éxito.

El presente trabajo de investigación se basó formular un modelo ensamblado robusto que determine el estado final de un proyecto, agregando un nuevo enfoque: basarse solamente en proyectos de tecnología, la segunda categoría con más baja probabilidad de éxito al final de una campaña. En estudios previos, los modelos planteados resultaron bastante aceptables debido a que el resto de categorías de proyectos balancearon la inequidad existente en las dos clases del estado.

El reto principal fue el de construir modelos predictivos que consideren las tres características más importantes de un proyecto: la primera basada en la metainformación, la segunda, en la imagen principal del proyecto y la tercera, en la descripción del mismo, para ser ensamblados más adelante en uno solo.

Para ello, se recolectó un total de 27,251 proyectos tecnológicos en Kickstarter entre los períodos 2009-2019, de los cuales 27,035 registros finalmente fueron usados para cada uno de los tres modelos. Algunos proyectos provenientes de países fuera del territorio de los Estados Unidos y en distintos idiomas fueron considerados dentro de esta cantidad ya que no afectó al rendimiento general como en casos particulares de algunos estudios previos.

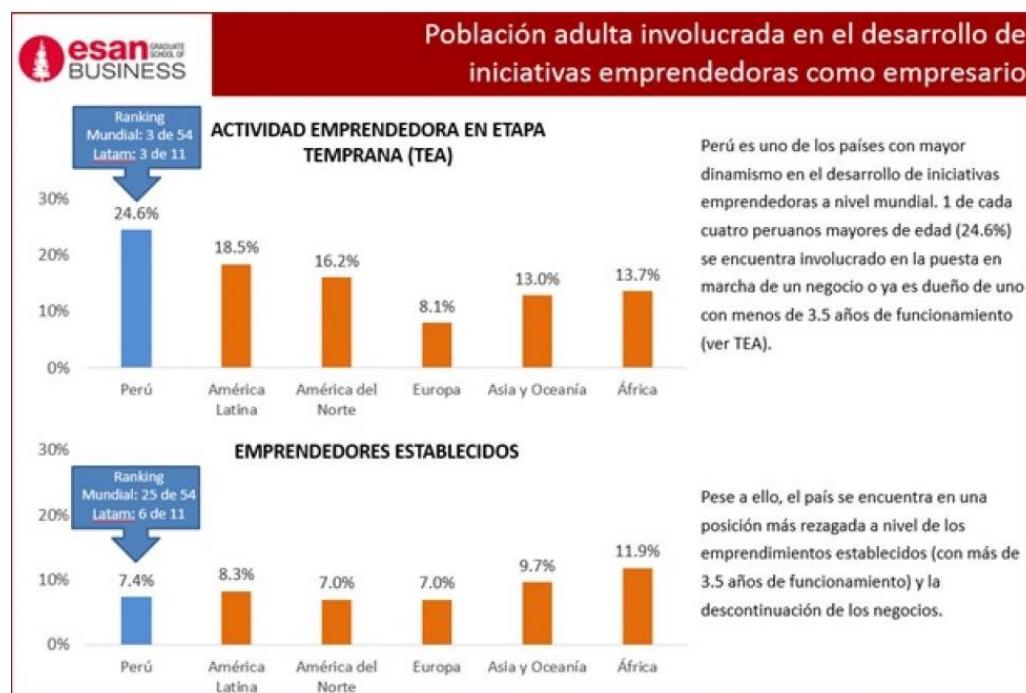
La principal motivación de este trabajo fue la de aportar una herramienta de ayuda para los emprendedores que les permita estimar el estado final del financiamiento de su proyecto de tecnología, es decir, exitoso o fracasado, con un nivel confiable de probabilidad de éxito del mismo durante el transcurso de su campaña, permitiendo además servir de soporte en la toma de decisiones de cara a lograr su principal objetivo.

## 1. CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

### 1.1. Descripción de la Realidad Problemática

El emprendimiento hoy en día es una realidad en todo el mundo. Desde crear productos nuevos hasta crear nuevas formas de hacer las cosas, todo gracias a ideas nacidas a partir de querer satisfacer nuestras propias necesidades.

Nuestro país no es ajeno a ello. El 50.6% de la población entre 18 y 64 años tiene la expectativa de iniciar un emprendimiento dentro de los tres próximos años de acuerdo al último reporte de Global Entrepreneurship Monitor (GEM) 2014. El 62.3% de la población entre ese rango de edad, además, tiende a ser más optimista en su percepción de oportunidades. Asimismo, según informa la Cámara de Comercio de Lima, la iniciativa emprendedora responde más a la identificación de una oportunidad de negocio que a una falta de oportunidad de empleo (Redacción Gestión, 2015). Sin embargo, en un estudio más reciente basado en una encuesta realizada a residentes peruanos entre junio y julio del 2017 desarrollada por el equipo GEM Perú y ESAN a 2080 personas entre el mismo rango de edad, el 24.6% de emprendimientos se encuentra en fase temprana, es decir, representa una dificultad para el emprendedor peruano llegar a etapas más avanzadas como un emprendimiento establecido (negocios con más de 3.5 años, que representan solo el 7.4% para Perú), ubicando así a nuestro país en la posición 25 de 54 economías a nivel mundial (Redacción Gestión, 2018). En la **Figura 1** se aprecian algunas ratios del estudio.



**Figura 1.** Resultados y ratios obtenidos en la encuesta por GEM y ESAN.

**Fuente:** (Redacción Gestión, 2018).

Estos resultados desfavorables tienen como base el ecosistema poco beneficioso para los emprendimientos que permitan su establecimiento en el entorno nacional, con condiciones asociadas al acceso de financiamiento, políticas gubernamentales que alienen la implementación de Innovación y Desarrollo en las empresas, acceso a infraestructura física y asesoría a nivel comercial y profesional, como sostiene el investigador del equipo GEM Perú Carlos Guerrero (Redacción Gestión, 2018). La Asociación de Emprendedores de Perú (ASEP) afirma, asimismo, que en la región solo se invierte el 1.5% del PIB en actividades de ciencia, tecnología e innovación, y las limitaciones son dadas por barreras burocráticas ejercidas por el Gobierno y el sector privado (Asociación de Emprendedores de Perú, 2018). En adición a esto, otras razones que representan barreras para emprender son la falta de conocimientos en la iniciación de un negocio, su tramitación, la fuente de financiamiento del proyecto o búsqueda de inversionistas, la cultura, la falta de fomento de emprendimiento y la falta de una red de contactos (Sandoval).

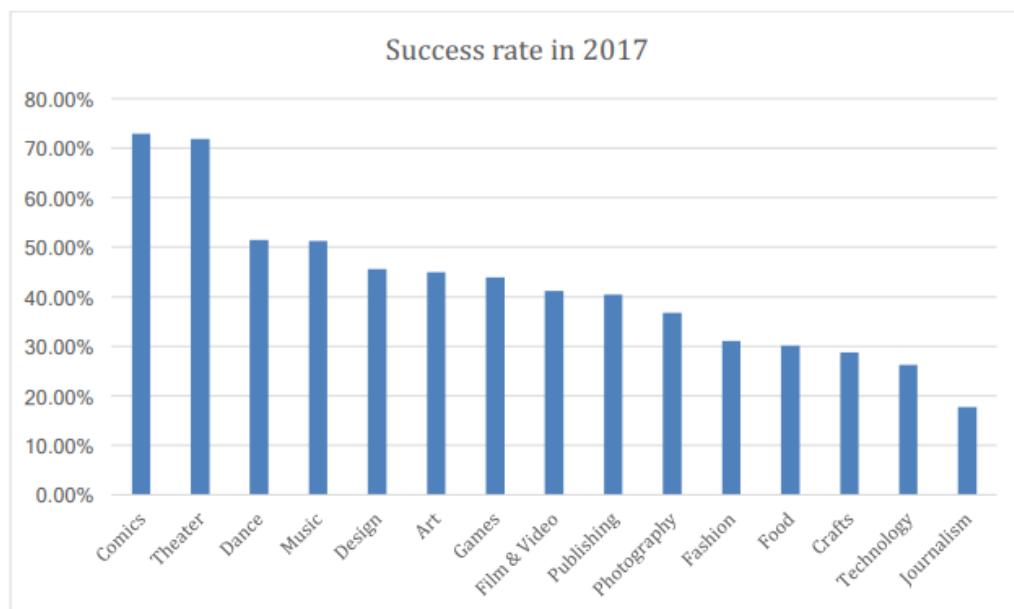
Ante estas limitaciones, en la actualidad muchos emprendedores se ven forzados a mostrar sus proyectos al público en la Internet con el fin de captar personas interesadas en ayudarlos en el financiamiento de estos. Por ello, se han creado plataformas web con el fin de permitir la interacción entre los proyectos publicados en un determinado tiempo, el cual puede variar entre 30 y 120 días, y la comunidad en general que deseé colaborar con una cantidad de dinero para su financiamiento. El sitio web solo servirá para mostrar los proyectos presentados a detalle por los creadores y la promoción de estos al público. La idea es que, al término de este plazo de tiempo, el proyecto sea financiado y se logre convertir en una realidad. A esta práctica se le conoce como crowdfunding (Universo Crowdfunding, s.f.).

En Latinoamérica, son muy pocos los países los que se incorporan en el crowdfunding, tales como Chile, México, Argentina y Brasil. Sin embargo, el modelo funciona distinto a países de Norteamérica y Europa debido a la cultura diferente y resistencia a su implementación por la poca confianza en el éxito de los proyectos. En los últimos años se decidió seguir una manera muy similar a los modelos de Estados Unidos, basados en la creación de campañas de un emprendedor para obtener fondos para sus ideas con la moneda norteamericana pero limitados a las leyes económicas de cada país (Solidaridad Latina).

Entre los sitios web más conocidos de crowdfunding están Kickstarter e Indiegogo. Kickstarter, desde su inicio en 2009, es una plataforma de financiamiento de proyectos creativos de todo tipo, los cuales incluyen películas, juegos, música, arte, diseño y tecnología. Actualmente, se han registrado más de 162 mil proyectos realizados, 16

millones de contribuyentes y 4,3 miles de millones de dólares fondeados (Kickstarter). La plataforma utiliza un modelo de financiamiento llamado “todo o nada”, el cual consiste en que si un proyecto no alcanza su meta de financiamiento en un determinado plazo de tiempo, no se realiza ninguna transacción de fondos (Kickstarter). Si bien los patrocinadores apoyan estos proyectos por motivos personales y distintos para hacerlos realidad, ellos no obtienen la propiedad o los ingresos de los proyectos que financian, sino que los creadores conservan la totalidad de su trabajo (Kickstarter).

Para los proyectos tecnológicos, en contraste, la ratio de éxito es uno de los más bajos de las categorías existentes junto con Artesanía (28.71%) y Periodismo (17.78%), como se aprecia en la **Figura 2**.



**Figura 2.** Ratio de éxito de proyectos en Kickstarter (2017).

**Fuente:** (Zhou H. , 2017).

Ya existen estudios previos para predecir la probabilidad de éxito de financiamiento para este tipo de proyectos utilizando técnicas de Aprendizaje Automático. Sin embargo, la mayoría de los modelos predictivos propuestos no arrojan resultados con exactitud muy alta ya que su rango varía entre 60 y 70%. Esto conlleva a generar imprecisión para pronosticar confiablemente el éxito de financiamiento de estos proyectos de tecnología. Para el presente trabajo de tesis, se creó un modelo predictivo alimentado de datos históricos de la plataforma para estimar el estado final de financiamiento de un proyecto aleatorio, así como su probabilidad de éxito.

## 1.2. Formulación del Problema

### 1.2.1. Problema General

Carencia de modelos predictivos usando técnicas de Aprendizaje Automático para proyectos tecnológicos con niveles aceptables de precisión (mayor a 0.70).

### 1.2.2. Problemas Específicos

- **P1:** Variables de proyectos no normalizadas y varianzas altas.
- **P2:** Datos faltantes o incompletos de proyectos.
- **P3:** Parámetros de modelos no ajustados.
- **P4:** Sobreajuste de aprendizaje de modelos y clasificación incorrecta de las dos clases del estado final de financiamiento (exitoso o fracasado).
- **P5:** Predicción incorrecta de estado de financiamiento de un proyecto tecnológico.

## 1.3. Objetivos de la Investigación

### 1.3.1. Objetivo General

Construir modelo(s) predictivo(s) usando técnicas de Aprendizaje Automático para proyectos tecnológicos con nivel de precisión aceptable (mayor a 0.70) a partir de la metainformación, imagen y/o descripción del proyecto.

### 1.3.2. Objetivos Específicos

- **O1:** Normalizar variables de proyectos y reducir niveles altos de varianza.
- **O2:** Eliminar datos faltantes o incompletos de proyectos.
- **O3:** Ajustar parámetros de modelos.
- **O4:** Evitar sobreajuste de aprendizaje de modelos para clasificar correctamente las dos clases del estado final de financiamiento (exitoso o fracasado).
- **O5:** Predecir correctamente el estado final de financiamiento de cualquier proyecto tecnológico.

## **1.4. Justificación de la Investigación**

### **1.4.1. Teórica**

El presente trabajo de tesis basado en predecir el éxito de financiamiento de un proyecto tecnológico en Kickstarter busca crear un modelo predictivo robusto que sea aplicable a cualquier tipo de proyecto en esta plataforma con mayor exactitud y precisión que los existentes y desarrollados en estudios pasados de los cuales se toman como referencias con la selección y posterior implementación de las técnicas más adecuadas de Aprendizaje Automático, incluso por aquellas no consideradas anteriormente, esto con el fin de ayudar en la toma de decisiones para las campañas dadas en el contexto descrito.

### **1.4.2. Práctica**

El presente trabajo de tesis se realiza porque actualmente existe la necesidad por parte de emprendedores, sean empresas o personas, patrocinadores y personas interesadas de determinar la viabilidad de uno o más proyectos, conocer el nivel de aceptación al lanzar un producto en el mercado y otros tipo de decisiones en los negocios que implican conocer el comportamiento de los consumidores y el público en general con la implementación de técnicas de Aprendizaje Automático debido a su uso común y frecuente. Por ello, se busca crear y utilizar un modelo predictivo para que a partir del pronóstico de éxito de financiamiento de un proyecto tecnológico en Kickstarter ayude en la toma de decisiones de los responsables de la campaña durante el tiempo en que esta se encuentre aún en proceso con el fin de potenciar las estrategias de promoción del proyecto o buscar alternativas en caso sea desfavorable.

### **1.4.3. Metodológica**

La razón principal es incrementar la precisión en los pronósticos de éxito de financiamiento de un proyecto tecnológico en Kickstarter con el fin de lograr disminuir la incertidumbre existente al momento de apostar por estos ya que actualmente son altamente riesgosos. Para ello, se abordarán conceptos de Aprendizaje Automático que determinen las herramientas y métodos necesarios para la creación de un modelo predictivo, empezando con el entendimiento de los datos y, de ser necesario, efectuar limpieza de datos. Asimismo, durante el modelado y previo al despliegue se procederá a probar el modelo final esperado varias veces con el fin de mejorar su performance y exactitud para obtener resultados altamente precisos y confiables.

## 1.5. Delimitación del Estudio

### 1.5.1. Espacial

El presente trabajo de tesis comprende el territorio de los Estados Unidos ya que tanto la campaña del proyecto a servir para la investigación como los datos fuentes de proyectos relacionados financiados previamente, que servirán para la elaboración del modelo predictivo, se encuentran en dicho país.

### 1.5.2. Temporal

El periodo de tiempo abarcará desde el año 2009, fecha en el cual se tiene registrado los primeros conjuntos de datos de proyectos en Kickstarter hasta el mes de agosto del año 2019, últimos registros descargados hasta el inicio del presente trabajo.

### 1.5.3. Conceptual

El trabajo de tesis consistirá en la implementación de un modelo predictivo de éxito de financiamiento de un proyecto tecnológico en Kickstarter basado en técnicas y conceptos de Aprendizaje Automático, previamente evaluando cuál de todas las existentes genera un mejor desempeño para su uso y análisis de resultados.

## 1.6. Hipótesis

### 1.6.1. Hipótesis General

El modelo predictivo creado con técnicas de Aprendizaje Automático logrará tener un nivel de precisión aceptable (mayor a 0.70) a partir de la metainformación, imagen y/o descripción del proyecto.

### 1.6.2. Hipótesis Específicas

- ✓ **H1:** Las variables de los proyectos descargados se normalizarán y se reducirán los niveles altos de varianza.
- ✓ **H2:** Los datos faltantes o incompletos de los proyectos serán eliminados.
- ✓ **H3:** Los parámetros de los modelos usados serán ajustados.
- ✓ **H4:** Se evitará el sobreajuste de aprendizaje de modelos para clasificar correctamente las dos clases del estado final de financiamiento.
- ✓ **H5:** El estado final de financiamiento de cualquier proyecto tecnológico será predicho correctamente.

### 1.6.3. Matriz de Consistencia

PROBLEMAS	OBJETIVOS	HIPÓTESIS	VARIABLES
<b>Problema General</b> Carencia de modelos predictivos usando técnicas de Aprendizaje Automático para proyectos tecnológicos con niveles aceptables de precisión (mayor a 0.70)	<b>Objetivo General</b> Construir modelo(s) predictivo(s) usando técnicas de Aprendizaje Automático para proyectos tecnológicos con nivel de precisión aceptable (mayor a 0.70) a partir de la metainformación, imagen y/o descripción del proyecto.	<b>Hipótesis General</b> El modelo predictivo creado con técnicas de Aprendizaje Automático logrará tener un nivel de precisión aceptable (mayor a 0.70) a partir de la metainformación, imagen y/o descripción del proyecto.	<b>Variable independiente</b> Estado final de financiamiento de un proyecto tecnológico en la plataforma Kickstarter.
<b>Problemas Específicos</b> <b>P1:</b> Variables de proyectos no normalizadas y varianzas altas.  <b>P2:</b> Datos faltantes o incompletos de proyectos.  <b>P3:</b> Parámetros de modelos no ajustados.  <b>P4:</b> Sobreajuste de aprendizaje de modelos y clasificación incorrecta de las dos clases del estado final de financiamiento (exitoso o fracasado).  <b>P5:</b> Predicción incorrecta de estado de financiamiento de un proyecto tecnológico.	<b>Objetivos Específicos</b> <b>O1:</b> Normalizar variables de proyectos y reducir niveles altos de varianza.  <b>O2:</b> Eliminar datos faltantes o incompletos de proyectos.  <b>O3:</b> Ajustar parámetros de modelos.  <b>O4:</b> Evitar sobreajuste de aprendizaje de modelos para clasificar correctamente las dos clases del estado final de financiamiento (exitoso o fracasado).  <b>O5:</b> Predecir correctamente el estado final de financiamiento de cualquier proyecto tecnológico.	<b>Hipótesis Específicas</b> <b>H1:</b> Las variables de los proyectos se normalizarán y se reducirán los niveles altos de varianza.  <b>H2:</b> Los datos faltantes o incompletos de los proyectos serán eliminados.  <b>H3:</b> Los parámetros de los modelos usados serán ajustados.  <b>H4:</b> Se evitará el sobreajuste de aprendizaje de modelos para clasificar correctamente las dos clases del estado final de financiamiento (exitoso o fracasado).  <b>H5:</b> El estado final de financiamiento de cualquier proyecto tecnológico será predicho correctamente.	<b>Variables dependientes</b> ✓ Meta de la campaña. ✓ Duración de la campaña. ✓ Número de patrocinadores. ✓ Monto total de promesas. ✓ Promesa por patrocinador. ✓ Fecha de lanzamiento del proyecto. ✓ Fecha de finalización del proyecto. ✓ Descripción del proyecto. ✓ Nombre del proyecto. ✓ Categoría del proyecto. ✓ Subcategoría del proyecto. ✓ Localización del proyecto. ✓ Actualizaciones. ✓ Comentarios. ✓ Número de amigos en Facebook del creador. ✓ Número de seguidores en Twitter del creador. ✓ Presencia de video en el proyecto. ✓ Presencia de imagen en el proyecto. ✓ Cantidad de tweets mencionando al proyecto.

**Tabla 1.** Matriz de Consistencia.

**Fuente:** Elaboración Propia

## 2. CAPÍTULO II: MARCO TEÓRICO

### 2.1. Antecedentes de la investigación

Al investigar trabajos relacionados previos para el estado de arte, se encontraron trabajos de investigación basados en la predicción de éxito o fracaso de campañas en Kickstarter o plataformas similares y el análisis de estas utilizando conjunto de datos de la propia plataforma o almacenadas en otros repositorios, con sus variables respectivas. En la mayoría de casos consideraron variables básicas que se obtienen del repositorio de las plataformas de crowdfunding, en otros casos consideraron variables cualitativas basadas en texto y descripción de proyectos, y en otros antecedentes usaron nuevas técnicas poco convencionales como el Aprendizaje Profundo e híbridos de modelos para obtener los mejores resultados posibles.

**PRIMER ANTECEDENTE:** “Supervised Learning Model for Kickstarter Campaigns with R Mining”, artículo de la revista científica International Journal of Information Technology, Modeling and Computing (Kamath & Kamat, 2018).

✓ **Realidad problemática:**

El crowdfunding es un paradigma talentoso usado en favor de los creadores de proyectos para solicitar fondos hacia cualquier persona interesada en realizar proyectos y cumplir el rol de patrocinador. Kickstarter actualmente es la plataforma web basado en crowdfunding más grande existente. Sin embargo, no todas las campañas que se encuentran en su portal son exitosas. Algunos trabajos previos citados en este artículo citan que el lenguaje usado en las campañas alcanza el poder predictivo de 58.86%, es decir, la descripción semántica de un proyecto ayuda considerablemente en la predicción de éxito debido a que muchos patrocinadores se fijan en la calidad presente en una campaña antes de invertir. Sin embargo, es dejada de lado en muchos trabajos de investigación destinados a la predicción de éxito de financiamiento, así como también las interacciones de un proyecto en redes sociales. Para el tiempo en que este artículo fue publicado, el nivel de exactitud de los modelos predictivos ya existentes bordeaba por el 68% como el descrito anteriormente. Los autores usan un conjunto de datos con más de 120 páginas de proyectos recolectadas en Kickstarter y otras fuentes secundarias, así como variables entre las que destacan el número de actualizaciones, el nivel de recompensa ofrecido, la duración, disponibilidad de video descriptivo del proyecto, entre otros.

✓ **Problema general:**

Modelos predictivos que no detectan patrones ocultos en la información de proyectos crowdfunding para su clasificación y con nivel de exactitud aún bajo (menos del 70%) para predecir con mayor precisión el éxito de su financiamiento.

✓ **Problemas específicos:**

- ¿Existe relevancia de las propiedades de los proyectos en el éxito de predicción?
- ¿Existen modelos de clasificación para campañas de Kickstarter?
- ¿Existe alguna variable muy influyente tanto en el éxito de la campaña de un proyecto como en la predicción que se le pueda aplicar posteriormente?
- ¿Se podrá diseñar un modelo que pueda adaptarse a la complejidad de un proyecto sin verse afectada su performance general?

✓ **Objetivo general:**

Desarrollar un sistema con técnicas de Aprendizaje Automático usando el entorno R aplicada a conjunto de datos de campañas en Kickstarter para clasificar proyectos entrenando diferentes clasificadores en esta data y predecir con un alto nivel de exactitud (aproximadamente 90%).

✓ **Objetivos específicos:**

- Determinar la relevancia de las propiedades de los proyectos en el éxito de predicción.
- Diseñar modelos diferentes de clasificación para campañas de Kickstarter basados en paquetes de R (exactitud del primer modelo, Naive Bayes: 84%, exactitud del segundo modelo, Red Neuronal: 94%, exactitud del tercer modelo, Random Forest: 78%, exactitud del cuarto modelo, Árboles de decisión: 52%).
- Determinar la existencia de alguna(s) variable(s) que influya considerablemente la performance de la campaña de un proyecto en Kickstarter.
- Diseñar un modelo de Aprendizaje Automático que pueda adaptarse a la complejidad de un proyecto, viéndose afectada su performance en la menor cantidad posible.

**SEGUNDO ANTECEDENTE:** “Predicting Success in Equity Crowdfunding”, artículo de la revista Joseph Wharton Scholars de la Universidad de Pensilvania (Beckwith, 2016).

✓ **Realidad problemática:**

El crowdfunding de inversión está haciendo cada vez más popular. Este concepto permite que los emprendedores ofrezcan algún tipo de producto o servicio como compensación por contribuciones financieras una vez que ya se tengan ventas reales

o acuerdos comerciales, a diferencia por ejemplo del crowdfunding de recompensas que ofrece algo a sus patrocinadores desde la concepción del proyecto. Resulta ser muy interesante para los patrocinadores ya que no se necesita contar con un capital muy alto. A cambio, ellos recibirán algo a cambio una vez que el proyecto se realice. Este factor permite su mayor probabilidad de éxito de financiamiento al momento de darse una campaña de este tipo de crowdfunding ya que los patrocinadores pueden invertir en más de un proyecto a la vez. A partir de este punto nace la siguiente cuestión: “Dados diferentes start-ups con similares características observables, ¿qué motiva a pequeños patrocinadores a invertir en ciertos start-ups y no en otros?” Para responder esta interrogante, se recolectó un conjunto de datos de 2,603 empresas en San Francisco, ciudad en donde fue realizada la investigación, además del análisis de variables relacionadas a lo anterior explicado como el número de menciones en la prensa acerca del proyecto, el número de fundadores del mismo, número de empleados, entre otros.

✓ **Problema general:**

¿Existe relación entre las características de una compañía determinada y su capacidad para recaudar fondos en una plataforma de financiamiento colectivo de capital?

✓ **Problemas específicos:**

- ¿Es posible predecir el éxito de una compañía que aplica crowdfunding de inversión a partir de sus características basadas en el proyecto, éxito en el pasado del creador en la plataforma y su actividad en redes sociales?
- ¿Es determinante en el éxito de financiamiento de un proyecto el hecho de que uno de los creadores tenga un nivel académico prestigioso?
- ¿Es determinante en el éxito de financiamiento de un proyecto alguna característica en particular de la compañía como por ejemplo el número de fundadores o cantidad de empleados?

✓ **Objetivo general:**

Determinar la relación entre las características de una compañía determinada y su capacidad para recaudar fondos en una plataforma de financiamiento colectivo de capital.

✓ **Objetivos específicos:**

- Estimar predicción de éxito de una compañía que aplica crowdfunding de inversión a partir de sus características basadas en el proyecto, éxito en el pasado

del creador en la plataforma y su actividad en redes sociales (exactitud del primer modelo, Regresión Logística: 87%, exactitud del segundo modelo, Árbol de Decisión CART: 85%, exactitud del tercer modelo, Naive Bayes: 83%, exactitud del cuarto modelo, Máquina de Vectores de Soporte: 86%).

- Determinar la relación entre el éxito de financiamiento de un proyecto y el probable nivel académico prestigioso de uno de sus creadores con herramientas de Aprendizaje Automático para su performance en el modelo predictivo.
- Determinar la relación entre el éxito de financiamiento de un proyecto y alguna característica en particular como por ejemplo el número de fundadores o cantidad de empleados con herramientas de Aprendizaje Automático para su performance en el modelo predictivo.

**TERCER ANTECEDENTE:** “Money Talks: A predictive Model on Crowdfunding Success Using Project Description”, resumen de la conferencia Twenty-first Americas Conference on Information Systems en Puerto Rico (Zhou, y otros, 2015).

✓ **Realidad problemática:**

Las investigaciones existentes de crowdfunding se centran principalmente en las variables básicas del proyecto como la categoría y la meta; sin embargo, son muy pocos estudios basados en el contenido de la información, es decir, en la descripción del mismo. Los autores de este paper hacen hincapié en que la importancia de la descripción del proyecto se basa justamente en la gran asociación que tiene con el éxito de la financiación debido a la información divulgada del proyecto hacia cualquier tipo de público, así como la propensión por parte de los dueños del proyecto a usarla estratégicamente como una herramienta de marketing para influir en las decisiones de contribución de los posibles patrocinadores. Para la comprensión del proceso de esta influencia, teóricamente puede verse apoyada por el Modelo de Probabilidad de Elaboración (ELM), anteriormente usada en contextos de investigación relacionados a la psicología social, mercadotecnia y literatura de investigación del consumidor. En la presente acta de conferencia, los autores implementaron la regresión logística en un conjunto de datos de 154,561 proyectos usando como variables el contenido de información como por ejemplo la descripción y antecedentes de proyectos financiados previamente por ellos, además de apoyarse por “variables tradicionales” en este tipo de contexto como la meta, duración,

categoría, e información basada en el uso de redes sociales por parte de los dueños del proyecto.

✓ **Problema general:**

¿Existen suficientes trabajos de investigación sobre modelos predictivos de éxito de financiamiento de crowdfunding basadas en el uso de la descripción del proyecto?

✓ **Problemas específicos:**

- ¿Es posible lograr una mejor exactitud en el modelo predictivo que los de trabajos previos a partir del uso de la descripción del proyecto?
- ¿Se puede implementar distintos modelos predictivos usando la descripción del proyecto?
- ¿Es posible reducir la brecha en la investigación de predicción de éxito de crowdfunding que implementan variables basadas en el contenido de la información de un proyecto?

✓ **Objetivo general:**

Desarrollar un modelo predictivo de éxito de crowdfunding usando la descripción del proyecto.

✓ **Objetivos específicos:**

- Lograr una mejor exactitud en el modelo predictivo que los de trabajos previos a partir del uso de la descripción del proyecto (modelo propuesto: 73%).
- Implementar distintos modelos de predicción usando la descripción del proyecto (exactitud del primer modelo: 58%, exactitud del segundo modelo: 69%).
- Reducir la brecha en la investigación de predicción de éxito de crowdfunding con el uso de variables basadas en el contenido de la información de un proyecto.

**CUARTO ANTECEDENTE:** “The Determinants of Crowdfunding Success: A Semantic Text Analytics Approach”, artículo de la revista científica Decision Support Systems (Yuan, Lau, & Xu, 2016).

✓ **Realidad problemática:**

En la era de la web social, el crowdfunding se ha convertido en una forma cada vez más importante para que empresarios o pequeñas empresas consigan capitales de las masas para apoyar sus proyectos. Algunos de sitios web de crowdfunding como Kickstarter e IndieGoGo actúan como intermediarios en países desarrollados para hacer llegar los proyectos a una gran cantidad de público con el fin de conseguir inversionistas. Sin embargo, el porcentaje de proyectos de crowdfunding que alcanzan

sus metas es muy reducido, tan solo el 44% de ellos lo logran en Kickstarter. Investigaciones anteriores estudiaron casos de predicción de crowdfunding en modelos basados en recompensas y análisis de características poco profundas del lenguaje en descripción de los proyectos (por ejemplo, número de palabras, errores ortográficos, etc) así como en la implementación de métodos de Aprendizaje Automático. El factor diferencial de este paper se basó en el estudio de modelos de crowdfunding no basados en recompensas, así como el uso de métodos de modelado de temas para extraer características tópicas de descripciones de proyectos (por ejemplo, semántica latente) y descripciones de recompensas para predecir el éxito de la recaudación de fondos en dos conjuntos de datos con 1,000 proyectos en total.

**✓ Problema general:**

¿Es posible crear un nuevo marco basado en análisis de texto que puede extraer la semántica latente de las descripciones textuales de los proyectos?

**✓ Problemas específicos:**

- ¿Es posible construir una red semántica entre palabras para guiar el proceso de modelado de temas?
- ¿Es posible analizar casos de proyectos de crowdfunding cuyos modelos no solamente sean basados en recompensas?
- ¿Es posible analizar casos de éxito de crowdfunding que no sean basados en plataformas de crowdfunding en el idioma inglés?
- ¿Es posible elaborar distintos modelos de Aprendizaje Automático con dos conjuntos de datos diferentes?
- ¿Es posible evaluar resultados de modelos de Aprendizaje Automático con dos conjuntos de datos diferentes aplicando dos métricas distintas?

**✓ Objetivo general:**

Diseñar un nuevo marco (DC-LDA) basado en análisis de texto que puede extraer la semántica latente de las descripciones textuales de los proyectos para predecir los resultados de la recaudación de fondos de estos proyectos.

**✓ Objetivos específicos:**

- Construir red semántica entre palabras para guiar el proceso de modelado de temas mediante estadísticas de co-ocurrencia de cada par.
- Analizar casos de proyectos de crowdfunding de modelos basados en recompensas y de no recompensas.
- Analizar éxito de crowdfunding basado en plataformas de crowdfunding chinas.

- Elaborar distintos modelos de Aprendizaje Automático (Random Forest, Red Neuronal de Propagación Inversa, Máquina de Vectores de Soporte, Máquina de Aprendizaje Extremo) con dos conjuntos de datos diferentes.
- Evaluar resultados de modelos de Aprendizaje Automático con dos conjuntos de datos diferentes aplicando dos métricas distintas (exactitud, mejor resultado: 78.5%; y precisión, mejor resultado: 77%).

**QUINTO ANTECEDENTE:** “Will your Project get the Green light? Predicting the success of crowdfunding campaigns”, resumen de la conferencia Pacific Asia Conference on Information Systems (PACIS) 2015 en New York, Estados Unidos (Chen, Chen, Chen, Yang, & Lin, 2015).

✓ **Realidad problemática:**

Después de surgir en el 2000, el crowdfunding se convirtió gradualmente en uno de los recursos de recaudación de fondos más populares. Sin embargo, el mecanismo para lograr el financiamiento solo se obtiene si la cantidad de dinero que se tiene como meta se alcanza en el plazo meta, resultando en un factor clave para determinar el éxito de la campaña de un proyecto. Los autores del paper proponen resolver el problema de predicción sobre una base de un conjunto de datos de alrededor de 4,000 proyectos con un modelo de distintas características que se obtienen de estos.

✓ **Problema general:**

¿Es posible desarrollar una técnica efectiva para predecir si una campaña será exitosa o no en diferentes momentos de tiempo de las campañas de crowdfunding?

✓ **Problemas específicos:**

- ¿Es posible desarrollar distintos modelos de clasificación basados en características y estáticas?
- ¿Es posible evaluar la performance de los modelos de clasificación construidos sobre éxito de campañas crowdfunding en diferentes momentos de tiempo de la campaña con diferentes métricas de clasificación?

✓ **Objetivo general:**

Desarrollar una técnica efectiva para predecir si una campaña será exitosa o no durante basada en producir una serie de modelos en diferentes momentos de tiempo de las campañas de crowdfunding.

✓ **Objetivos específicos:**

- Desarrollar distintos modelos de clasificación basadas en características estáticas (intrínsecas, mecanismo financiero, sentimiento y calidad de contenido) y dinámicas (interacción social y efecto progresión), así como modelo propuesta que incluya el conjunto de todas las características.
- Evaluar performance de modelos de clasificación sobre éxito de campañas crowdfunding en diferentes momentos de tiempo de la campaña (desde el día 0 hasta el día 7) con diferentes métricas de clasificación (precisión, mejor resultado: 86.69% en día 6; sensibilidad, mejor resultado: 88.43% en día 6; puntuación F1, mejor resultado: 87.55% en día 6; exactitud, mejor resultado: 89.72% en día 6).

**SEXTO ANTECEDENTE:** “Project Success Prediction in Crowdfunding Environments”, resumen de la conferencia WSDM’ 16 Proceedings of the Ninth ACM International Conference on web search and data mining en San Francisco (Li, Rakesh, & Reddy, 2016).

✓ **Realidad problemática:**

Durante los últimos años, los sitios web de crowdfunding ayudaron a empresas e individuos de todo el mundo a recaudar \$ 89 millones en 2010 y creció de manera explosiva a \$ 5.1 billones en 2013, convirtiéndose así en una alternativa viable para las personas que buscan la ayuda de bancos, corredores y otros intermediarios financieros para impulsar sus negocios. A pesar del tremendo éxito del crowdfunding, las estadísticas muestran que solo alrededor del 40% de los proyectos tienen éxito al cumplir su objetivo. Durante la última década se han elaborado una serie de modelos de clasificación que permitan pronosticar con cierto nivel de exactitud si algunos de estos proyectos tendrán éxito o no. Sin embargo, el hecho de estimar si esto ocurrirá durante el plazo dado de la campaña no puede proporcionar una guía adecuada a los patrocinadores que desean invertir en proyectos populares. El presente paper, además de ilustrar la debilidad de los enfoques basados en la clasificación como métodos de recomendación para los inversores, se basa en clasificar los más de 18,000 proyectos obtenidos según su fecha de éxito esperada para que los inversores puedan elegir algunos proyectos interesantes del grupo de proyectos altamente clasificados considerando además actividades promocionales en las redes sociales y otro conjunto de características obtenidas.

✓ **Problema general:**

¿Es posible predecir el éxito de campañas en Kickstarter considerando características obtenidas durante la operación de la campaña?

✓ **Problemas específicos:**

- ¿Es posible formular un modelo predictivo de éxito del proyecto que tenga mejor desempeño que los modelos de clasificación existentes?
- ¿Es posible demostrar que modelos ajustados por distribuciones logísticas y log-logísticas ejercen mejores performances que otros modelos de regresión mencionados en la literatura?
- ¿Es posible evaluar el conjunto de características más óptimas que se deben extraer del conjunto de datos de Kickstarter para predecir el éxito del proyecto?
- ¿Es posible demostrar que agregando algunas características temporales obtenidas durante la operación de la campaña del proyecto se puede mejorar dramáticamente el desempeño de la predicción?
- ¿Es posible comparar los cuatro modelos formulados con diferentes técnicas de Minería de datos usando como métrica el área bajo la curva (AUC)?

✓ **Objetivo general:**

Predecir el éxito de campañas en Kickstarter considerando características obtenidas durante la operación de la campaña.

✓ **Objetivos específicos:**

- Formular un modelo de predicción del éxito del proyecto aprovechando por completo los proyectos exitosos y fracasados durante la fase de entrenamiento para obtener un desempeño significativamente mejor que aquellos modelos que solo abarcan información de proyectos exitosos.
- Comparar modelos basados en distribuciones logísticas y log-logísticas, que representan una opción natural para ajustar los modelos paramétricos para datos de crowdfunding, con otros modelos de regresión censurados (que no incluyen proyectos fracasados) disponibles en la literatura.
- Evaluar el conjunto de características más óptimas que deben extraerse del conjunto de datos de Kickstarter para predecir el éxito del proyecto mediante la comparación de cuatro modelos de clasificación (modelo 1: características estáticas; modelo 2: características estáticas + características obtenidas de Twitter; modelo 3: características estáticas + características obtenidas luego de 3 días; modelo 4: características estáticas + características obtenidas luego de 3 días + características obtenidas de Twitter).

- Demostrar que agregando algunas características temporales obtenidas durante la operación de la campaña del proyecto (como luego de 3 días o de redes sociales) se puede mejorar dramáticamente el desempeño de la predicción.
- Comparar los cuatro modelos formulados con diferentes técnicas de Minería de datos (modelo Cox, regresión Tobit, regresión Buckley-James, índice de concordancia Boosting, regresión logística, regresión log-logística) usando como métrica el área bajo la curva (AUC).

**SÉPTIMO ANTECEDENTE:** “Effect of Social Media Connectivity on Success of Crowdfunding Campaigns”, artículo de la revista científica Procedia Computer Science (Kaur & Gera, 2017).

✓ **Realidad problemática:**

En los últimos años, el número de plataformas de crowdfunding y proyectos lanzados en ellas han aumentado de manera exponencial, pero el número de proyectos exitosos se está reduciendo. Sin embargo, muchos proyectos tienen potencial para recaudar fondos con éxito, pero fracasan debido a la falta de reconocimiento, publicidad y promoción adecuados. Los medios sociales desempeñan un papel importante en la difusión de palabras sobre una campaña y en la obtención de fondos con éxito. Después de lanzarse una campaña, su éxito se define por la interacción social de los creadores en la plataforma y en las redes sociales. Además, el tamaño de la red social del creador y su presencia en línea motiva a los patrocinadores a participar y financiar el proyecto. Para responder a la pregunta general, los autores desarrollaron un modelo predictivo (con una exactitud promedio de éxito de 76.7%) que comprenda, en un conjunto de datos de más de 4,000 proyectos, las características de la campaña y sumado a eso, recuperaron información sobre la conectividad con las diversas redes sociales como Facebook y Twitter, en este último se analizaron los tweets.

✓ **Problema general:**

¿Cuál es el impacto de la conectividad con las redes sociales y las interacciones sociales en el rendimiento del crowdfunding?

✓ **Problemas específicos:**

- ¿Es posible evaluar la performance de las variables del modelo con distintos modelos de Aprendizaje Automático?
- ¿Es posible evaluar la performance de la clasificación del modelo propuesto con diferentes métricas?

- ¿Es posible afirmar la relevancia de todas las variables del modelo para determinar el éxito de proyectos?

✓ **Objetivo general:**

Promover las campañas de crowdfunding mediante el uso de interacciones de los dueños del proyecto en redes sociales y estudiar su impacto en el éxito de financiamiento de proyectos.

✓ **Objetivos específicos:**

- Evaluar y comparar la performance de las variables del modelo con distintos modelos de Aprendizaje Automático (Naive Bayes, J48, Random Forest y regresión logística).
- Evaluar y comparar la performance de la clasificación del modelo propuesto (regresión logística, exactitud de 76.7%) con diferentes métricas (ratio Verdaderos Positivos, ratio Falsos Positivos, precisión, sensibilidad, métrica F, MCC, área bajo la curva ROC, área bajo la curva PRC).
- Evaluar la correlación entre el éxito y las variables para determinar su relevancia y performance en el modelo que ayude a la predicción de éxito de proyectos.

**OCTAVO ANTECEDENTE:** “Prediction of Crowdfunding Project Success with Deep Learning”, resumen de la conferencia 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE) del 12 al 14 de octubre del 2018 en Xi'an, China (Yu, y otros, 2018).

✓ **Realidad problemática:**

El crowdfunding ofrece una alternativa tanto para los creadores para vender sus productos como para los patrocinadores para invertir en negocios creativos. En Kickstarter, el número de proyectos ha desarrollado 85 veces más desde 2009 hasta 2015 y el monto total recaudado se ha incrementado de \$ 1.6 millones a más de \$ 600 millones de dólares en el mismo periodo. Sin embargo, el análisis empírico muestra que solo un tercio de las campañas de crowdfunding podrían cumplir su objetivo de recaudación de fondos. Los autores de este paper desarrollaron un modelo que predice el éxito del proyecto de crowdfunding en un conjunto de datos de más de 370,000 proyectos recuperados de Kaggle entre los periodos de marzo 2009 y marzo 2018 usando Aprendizaje Profundo (Deep Learning) en vez de técnicas tradicionales de Aprendizaje Automático (Machine Learning).

✓ **Problema general:**

¿Es posible desarrollar un modelo de predicción de éxito de proyectos en Kickstarter utilizando técnicas distintas al Aprendizaje Automático?

✓ **Problemas específicos:**

- ¿Qué criterios se deben seguir para construir un modelo de Aprendizaje Profundo basado en el Perceptrón Multicapa?
- ¿Es posible obtener mejores resultados en el modelo propuesto de Aprendizaje Profundo al comparar con modelos de Aprendizaje Automático?
- ¿Cómo se verá afectada la performance del modelo al incrementarse el volumen del conjunto de datos actual?

✓ **Objetivo general:**

Desarrollar un modelo de predicción de éxito de proyectos en Kickstarter utilizando Aprendizaje Profundo.

✓ **Objetivos específicos:**

- Construir modelo de Aprendizaje Profundo basado en el Perceptrón Multicapa definiendo los mejores parámetros (número de capas de entrada, número de neuronas capas ocultas, número de lotes, tipos de optimizador).
- Desarrollar y comparar performance de modelos de Aprendizaje Automático (Random Forest, Refuerzo Adaptativo-AdaBoost, Máquina de Vectores de Soporte-SVM, Árbol de Decisión, Regresión Logística, Naïve Bayes) con modelo propuesto de Aprendizaje Profundo (Perceptrón Multicapa).
- Realizar prueba de escalabilidad para determinar el impacto en la performance del modelo al incrementarse el volumen del conjunto de datos.

**NOVENO ANTECEDENTE:** “Estimating the Days to Success of Campaigns in Crowdfunding: A Deep Survival Perspective”, resumen de la conferencia The 33rd AAAI Conference on Artificial Intelligence (AAAI'2019) del 27 de enero al 1 de febrero del 2019 en Honolulu, Estados Unidos (Jin, Zhao, Chen, Liu, & Ge, 2019).

✓ **Realidad problemática:**

En los últimos años, el crowdfunding se ha convertido gradualmente en una forma popular para que los empresarios e individuos soliciten fondos del público para sus ideas creativas. Algunas de las plataformas más populares se encuentran Indiegogo. En esta área, algunos investigadores se enfocaron en predecir la tasa de éxito de las campañas. Sin embargo, los problemas de respaldar la predicción de la distribución y la predicción del tiempo de éxito de las campañas son más difíciles y complicados.

Los autores proponen elaborar un modelo de Aprendizaje Profundo considerando características usualmente poco tomadas en cuenta como el tiempo de éxito en un conjunto de datos de más de 14,000 campañas en Indiegogo.

✓ **Problema general:**

Los tiempos de éxito de las campañas crowdfunding son inobservables debido al alto nivel de fracaso (60%) en la financiación total de los proyectos.

✓ **Problemas específicos:**

- ¿Qué factores y características en el modelo impactan la performance de predicción de la campaña?
- ¿Cuál es el impacto de los comentarios de los patrocinadores en la campaña del proyecto?
- ¿Cómo se puede aprovechar el fuerte impacto que tiene la distribución de respaldo con la velocidad de recaudación de fondos?
- ¿Es posible obtener mejores resultados del modelo propuesto al evaluarlo y compararlo con modelos de Aprendizaje Automático?

✓ **Objetivo general:**

Desarrollar un modelo basado en Seq2seq con antecedentes de múltiples facetas (SMP) que permita integrar características heterogéneas para modelar conjuntamente las predicciones de la distribución de respaldo y del tiempo de éxito con alto nivel de exactitud (96% alcanzado en el mejor escenario).

✓ **Objetivos específicos:**

- Analizar características como la descripción de la campaña, información sobre beneficios, comentarios, entre otros, e integrarlas para lograr una predicción precisa.
- Modelar las relaciones entre los comentarios de los patrocinadores y las distribuciones de respaldo para determinar el nivel de impacto de estos en la campaña del proyecto.
- Predecir el tiempo de éxito de las campañas teniendo en cuenta la distribución de respaldo por su alto impacto en la velocidad de recaudación de fondos.
- Desarrollar modelos alternos de Aprendizaje Automático (COX, Tobit, regresión logística, regresión log-logística, Multitask L21, Multitask Lasso, BoostCOX, SurvivalSVM) y comparar performance y resultados con modelo propuesto.

**DÉCIMO ANTECEDENTE:** “Success Prediction on Crowdfunding with Multimodal Deep Learning”, resumen de la conferencia The Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19) del 10 al 16 de agosto del 2019 en Macao, China (Cheng, Tan, Hou, & Wei, 2019).

✓ **Realidad problemática:**

A pesar de que la información en un perfil de proyecto puede ser de diferentes modalidades, como texto, imágenes y metadatos, la mayoría de los enfoques de predicción existentes aprovechan solo la modalidad dominada por el texto. Hoy en día se han utilizado imágenes visuales ricas en más y más perfiles de proyectos para atraer patrocinadores, se ha realizado poco trabajo para evaluar sus efectos hacia la predicción del éxito. Además, la metainformación ha sido explotada en muchos enfoques existentes para mejorar la precisión de la predicción. Sin embargo, esta generalmente se limita a la dinámica después de la publicación de los proyectos, haciendo que tanto los creadores del proyecto como las plataformas no puedan predecir el resultado de manera oportuna.

✓ **Problema general:**

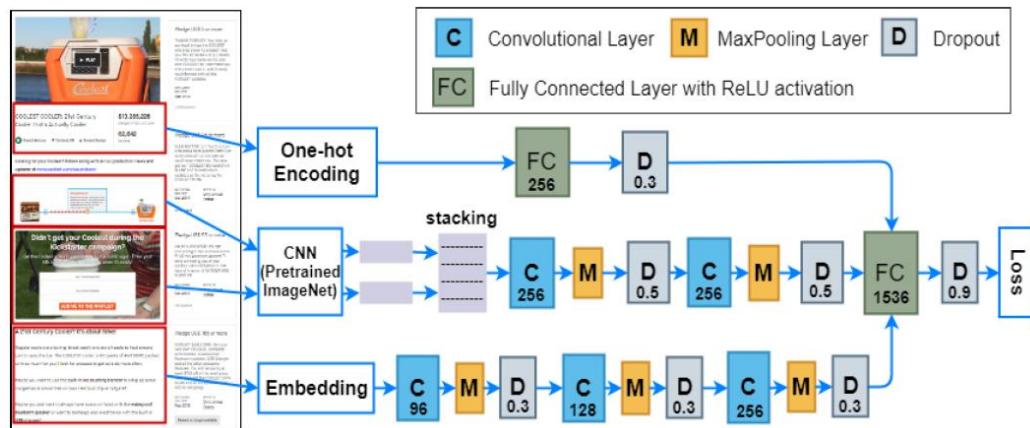
Carencia de estudios de predicción de éxito basados en la combinación de distintas modalidades de información, además de la textual, como la visual o metadatos que permitan crear un modelo más preciso que los antecedentes.

✓ **Problemas específicos:**

- ¿Será posible afirmar la obtención de mejores resultados que un modelo de Máquina de Vectores de Soporte (SVM) basado en los antecedentes, a partir del desarrollo de un modelo de Aprendizaje Profundo Multimodal (MDL) que contenga variables textuales y no textuales de proyectos en Kickstarter?
- ¿Será posible desarrollar modelos independientes de cada característica extraída (contenido textual, visual y metainformación) y comparar resultados de sus performances mediante distintas métricas?
- ¿Será posible determinar el impacto de la performance del modelo de Aprendizaje Profundo Multimodal al añadirle o restarle variables no textuales?

✓ **Objetivo general:**

Diseñar y evaluar esquemas avanzados de redes neuronales (ver **Figura 3**) que combinan información de diferentes modalidades para estudiar la influencia de interacciones sofisticadas entre texto, visual y metadatos en la predicción del éxito del proyecto en más de 20,000 proyectos obtenidos de Kickstarter.



**Figura 3.** Marco de trabajo representado en 3 partes: metainformación (parte superior), contenido visual (medio) y contenido textual (parte inferior).

**Fuente:** (Cheng, Tan, Hou, & Wei, 2019).

✓ **Objetivos específicos:**

- Desarrollar un modelo de Aprendizaje Profundo Multimodal (MDL) y comparar resultados (mejor resultado, AUC: 83.26%) con los de un modelo de Máquina de Vectores de Soporte (SVM) basado en los antecedentes (mejor resultado, precisión: 75.95%).
- Desarrollar modelos independientes (Bolsa de palabras para el contenido textual, VGG de 16 capas pre-entrenado para el contenido visual, y One-hot Encoding para la metainformación) de cada tipo de característica extraída (contenido textual, visual y metainformación) y comparar resultados de sus performances entre sí mediante distintas métricas.
- Determinar el impacto de la performance del modelo de Aprendizaje Profundo Multimodal al añadirle o restarle variables no textuales.

## 2.2. Bases Teóricas

### 2.2.1. Inteligencia Artificial

La Inteligencia Artificial es la inteligencia llevado a cabo por máquinas en las que una máquina “inteligente” ideal es un agente flexible que percibe su entorno y lleva a cabo acciones que maximicen sus posibilidades de éxito en algún objetivo (Poole, Mackworth, & Goebel, 1998). Este término se aplica cuando una máquina imita las funciones “cognitivas” que asocian los humanos con otras mentes (Russell & Norvig, Inteligencia Artificial: Un Enfoque Moderno, 2009).

Durante la historia de la humanidad, se han seguido 4 enfoques: dos centrados en el comportamiento humano y dos enfocados en torno a la racionalidad. El enfoque centrado en el comportamiento humano se basa en una ciencia empírica, es decir, mediante experimentos que incluyen hipótesis y confirmaciones. Este enfoque nace a partir de la prueba de Alan Turing, en 1950, en la cual, el célebre matemático inglés diseñó una prueba basada en la incapacidad de diferenciar entre entidades inteligentes indiscutibles y seres humanos por parte de un computador. Si este era capaz de diferenciar y superar la prueba mientras que el humano no, se afirma que se trataba de una “máquina inteligente”. Por ello, el computador debía contar con las siguientes capacidades: procesamiento de lenguaje natural para poder comunicarse, representación del conocimiento describiendo lo que percibe de su entorno, razonamiento automático utilizando la información procesada en su interior, y aprendizaje automático para adaptarse a nuevos eventos. Si el evaluador decide incluir una señal de video para evaluar la percepción de la computadora, se dice que se está realizando la Prueba Global de Turing. Para superarla, además de las 4 anteriormente mencionadas, la computadora debe contar además con las capacidades de visión computacional para percibir objetos y robótica con el fin de manipularlos. Todas estas seis capacidades o disciplinas abarcan la mayor parte de la Inteligencia Artificial (Russell & Norvig, Inteligencia Artificial: Un Enfoque Moderno, 2004).

Por el otro lado, el enfoque racional implica una combinación de ingeniería y matemáticas basándose en las “leyes del pensamiento”. Estas parten de la Grecia antigua, planteadas por grandes filósofos como Aristóteles en su intento de codificar la “manera correcta de pensar”, lo que más adelante derivó al estudio de la lógica. Más adelante, en el siglo XIX, se construyeron programas capaces de resolver problemas en notación lógica. De ahí que la tradición logista dentro del campo de la Inteligencia Artificial trata de construir sistemas inteligentes con estas capacidades. De todo lo

anterior dicho respecto al enfoque racional se creó el término de un agente racional, el cual actúa intentando lograr el mejor resultado, o de existir incertidumbre, el mejor resultado esperado. Finalmente, la amplia aplicación de la Inteligencia Artificial y sus fundamentos derivan en muchas ciencias de las cuales se pueden mencionar, además de la filosofía y las matemáticas, a la economía, neurociencia, psicología, la ingeniería computacional, la teoría de control y cibernetica, y hasta la lingüística (Russell & Norvig, *Inteligencia Artificial: Un Enfoque Moderno*, 2004).

Pero, ¿cómo es surge este amplio estudio de la Inteligencia Artificial? En 1943, basándose en la fisiología básica y funcionamiento de las neuronas en el cerebro, el análisis formal de la lógica proposicional de Russell y Whitehead, y la teoría computacional de Turing, dos estudiosos en neurociencia realizaron juntos el que sería considerado primer trabajo de Inteligencia Artificial. Warren McCulloch y Walter Pitts propusieron un modelo constituido por neuronas artificiales, en el que cada una de ellas se caracterizaba por estar “activada” o “desactivada”; la del primer tipo daba como resultado a la estimulación producida por una cantidad suficiente de neuronas vecinas. Como ejemplo, mostraron que cualquier función de cómputo podría calcularse mediante alguna red de neuronas interconectadas y que todos los conectores lógicos eran capaces de ser implementados usando estructuras sencillas de red. Seis años más adelante, Donald Hebb propuso una regla de actualización de intensidades de conexiones entre las neuronas, la que actualmente se le conoce como la “regla de aprendizaje Hebbiano” vigente hasta nuestros días. En 1956, Allen Newell y Herbert Simon inventaron un programa de computación en el taller de Dartmouth de John McCarthy, que era capaz de pensar de forma no numérica, basado en el Teórico Lógico, artículo que, además, fue rechazado de ser publicado en la revista *Journal of Symbolic Logic*. A pesar de ello, los trabajos de los colaboradores presentes en dicho taller se mantuvieron por 20 años más, siendo McCarthy quien acuñó el término de “Inteligencia Artificial” a este campo (Russell & Norvig, *Inteligencia Artificial: Un Enfoque Moderno*, 2004).

En la década de los años 80, la Inteligencia Artificial dio el gran salto de formar parte de la industria, en especial, de las compañías más grandes de los países desarrollados a través de grupos especializados para la realización de investigaciones de sistemas expertos, así como en la construcción de computadoras cada vez más potentes y capaces de resolver tareas más complejas.

Actualmente, la IA cuenta con muchas aplicaciones como la Minería de Datos, el procesamiento de lenguaje natural, la robótica, los videojuegos, entre otros. Dentro de ella se pueden encontrar otras ramas como por ejemplo el Aprendizaje Automático, Visión computacional, etcétera.

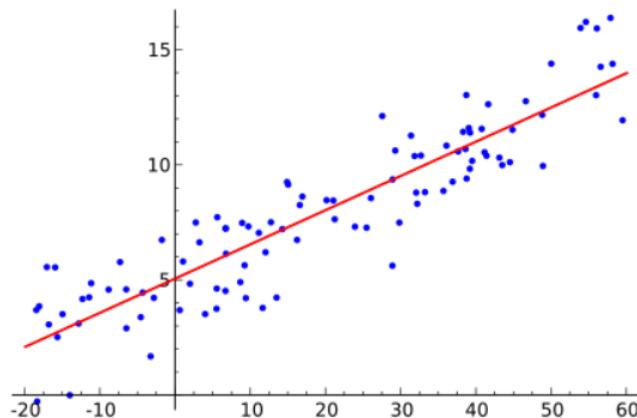
### 2.2.2. Aprendizaje Automático

El Aprendizaje Automático (*Machine Learning* por su nombre en inglés) es una rama de la Inteligencia Artificial cuyo fin es desarrollar técnicas que las computadoras pueden aprender a través de encontrar algoritmos y heurísticas que conviertan muestras de datos en programas sin necesidad de hacerlos (Russell & Norvig, Inteligencia Artificial: Un Enfoque Moderno, 2009). Sus algoritmos están compuestos por muchas tecnologías, como por ejemplo Aprendizaje Profundo, Redes Neuronales y Procesamiento de lenguaje natural, utilizadas en el aprendizaje supervisado y no supervisado, las cuales operan guiadas por lecciones de información existente (Gartner, s.f.). La premisa básica del aprendizaje automático es construir algoritmos que puedan recibir datos de entrada y usar análisis estadísticos para predecir una salida mientras se actualizan las salidas a medida que se dispone de nuevos datos (Alpaydin, 2014).

Como se mencionó, existen dos tipos de aprendizaje:

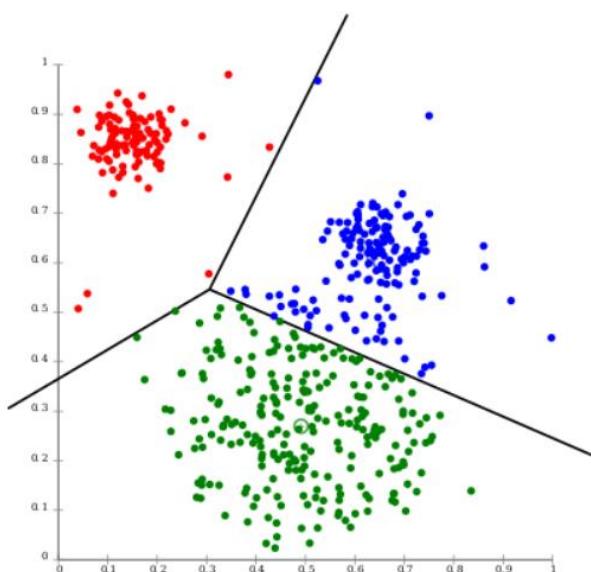
- ✓ **Aprendizaje Supervisado:** Se trabajan con datos etiquetados buscando obtener una función que asigne una respuesta de salida adecuada, denominadas etiquetas, a partir de unos datos de entrada denominadas características (Zambrano, 2018). Por lo general, los datos de entrada son conocidos como variables dependientes o X, mientras que los datos de salida son llamadas variables independientes o Y. Se le dice supervisado ya que el resultado depende de los datos que recibe de entrada, afectando su performance si estos son alterados.

Existen dos tipos de aprendizaje supervisado. El primero es la regresión, que consiste en obtener como resultado un número específico a partir de un conjunto de variables de las características, representado en la **Figura 4**; mientras que por otra parte está la clasificación, el cual se basa en encontrar distintos patrones ocultos para clasificar los elementos del conjunto de datos en diferentes grupos, como se aprecia en la **Figura 5** (Zambrano, 2018).



**Figura 4.** Ejemplo de algoritmo de regresión.

**Fuente:** (Zambrano, 2018).



**Figura 5.** Ejemplo de algoritmo de clasificación.

**Fuente:** (Zambrano, 2018)

Para el segundo tipo de aprendizaje supervisado, el algoritmo más usado es el de los K Vecinos más cercanos o *k-NN Nearest Neighbour* en inglés. Este se basa en la idea de que los nuevos ejemplos serán clasificados a la clase a la cual pertenezca la mayor cantidad de vecinos más cercanos del conjunto de entrenamiento más cercano a él. Sin embargo, el número k de vecinos más cercanos lo decide el usuario, de preferencia impar, para evitar ambigüedad al momento de clasificar un registro por parte del algoritmo (esto puede ocurrir por las mismas distancias existentes entre dos o más registros). Otra variante aplicada consiste en la ponderación de cada vecino de acuerdo a la distancia entre él y el ejemplar a ser clasificado, asignando mayor peso

a los más próximos (Sancho Caparrini, Clasificación Supervisada y No Supervisada, 2018). Por ejemplo, si  $x$  es el ejemplo que se desea clasificar,  $V$  son las posibles clases de clasificación, y  $\{x_i\}$  es el conjunto de los  $k$  ejemplos de entrenamiento más cercano, se define la siguiente fórmula:

$$w_i = \frac{1}{d(x, x_i)^2}$$

**Ecuación 1.** Cálculo de los pesos para el algoritmo K-NN mediante ponderación de sus distancias.

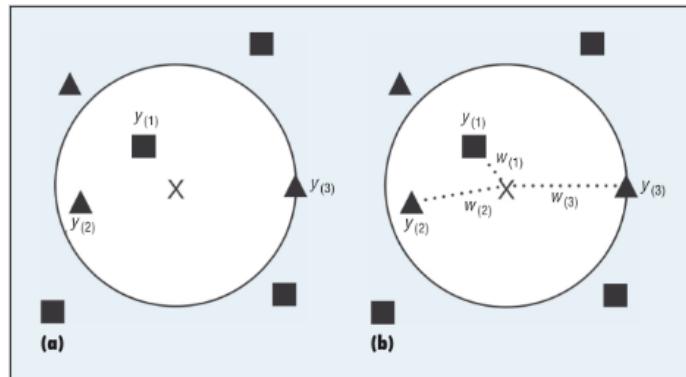
**Fuente:** (Sancho Caparrini, Clasificación Supervisada y No Supervisada, 2018).

y finalmente, la clase asignada a  $x$  es aquella que verifique que la suma de los pesos de sus representantes sea la máxima, representándose en la **Figura 6**:

$$\operatorname{argmax}_{v \in V} \sum_{i=1 \dots k, x_i \in v} w_i$$

**Ecuación 2.** Fórmula alternativa del algoritmo K-NN mediante sumatoria de pesos.

**Fuente:** (Sancho Caparrini, Clasificación Supervisada y No Supervisada, 2018).



**Figura 6.** Algoritmo de K Vecinos más cercanos con pesos ponderados.

**Fuente:** (Sancho Caparrini, Clasificación Supervisada y No Supervisada, 2018).

- ✓ **Aprendizaje No Supervisado:** A diferencia de la anterior, aquí se trabaja con datos no etiquetados para entrenar el modelo, ya que el fin es de carácter exploratorio y descriptivo de la estructura de los datos. No existen variables independientes o Y. La función es agrupar ejemplares, por lo que el algoritmo los cataloga por similitud en sus características y a partir de ahí, crea grupos o clústeres sin tener la capacidad de definir cómo es cada individualidad de cada uno de los integrantes de los mismos (Zambrano, 2018).

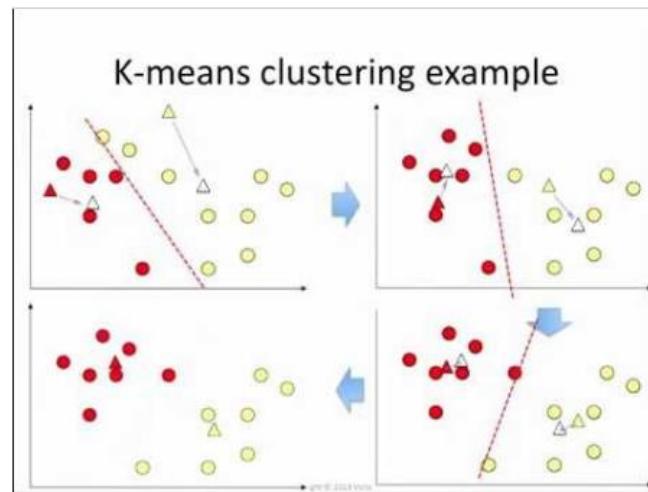
El algoritmo usado para este tipo de aprendizaje es el de las K medias o *k-means* en inglés. Este intenta encontrar una partición de las muestras en K agrupaciones, de manera que cada ejemplar pertenezca a una de ellas de acuerdo al centroide más cercano. Si bien el valor de K es definido por el usuario, a partir de pruebas de varias iteraciones se le puede consultar al algoritmo cuál es su valor óptimo. La intención es minimizar la varianza total del sistema. Por ejemplo, si se tiene el centroide  $c_i$  de la agrupación i-ésima, y  $\{x_j^i\}$  es el conjunto de ejemplos clasificados en esa agrupación, la función para lograr esto es la siguiente:

$$\sum_i \sum_j d(x_j^i, c_i)^2$$

**Ecuación 3.** Fórmula del algoritmo k-means.

**Fuente:** (Sancho Caparrini, Clasificación Supervisada y No Supervisada, 2018).

Representándose en la **Figura 7**, los pasos seguidos para este algoritmo comienzan con la selección de los K puntos como centros de los grupos. Luego, se asignarán los ejemplos al centro más cercano y se calculará el centroide de los ejemplos asociados a cada grupo. Finalmente, estos dos últimos pasos se repetirán hasta que ninguno de los centros pueda ser reasignados en las iteraciones.

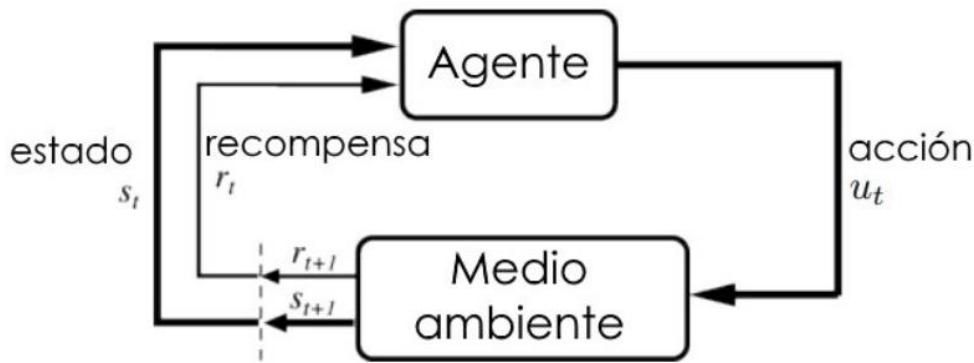


**Figura 7.** Funcionamiento del algoritmo de K medias.

**Fuente:** (Sancho Caparrini, Clasificación Supervisada y No Supervisada, 2018).

- ✓ **Aprendizaje por Refuerzo:** Se basa en que un agente racional puede tomar una decisión a partir de una retroalimentación llamada recompensa o refuerzo. A diferencia del Aprendizaje Supervisado, en donde el agente puede aprender solamente a partir de ejemplos dados, en este caso no basta solamente con proporcionárselos sino

también de “informarle” si lo está haciendo de la manera correcta o no. Por ejemplo, un agente que intenta aprender a jugar ajedrez necesita saber que algo bueno ha ocurrido cuando gana y algo malo ha ocurrido cuando pierde. La mejor recompensa que busca al finalizar el juego es vencer al oponente, y para ello debe estudiar todos los movimientos que este haga, la posición de las fichas en el tablero, entre otros. A este conjunto se le conoce como entorno o medio ambiente (Russell & Norvig, Inteligencia Artificial: Un Enfoque Moderno, 2004). Entonces, en resumen, representando en la **Figura 8**, y mencionando otro ejemplo, el aprendizaje por refuerzo está compuesto por un agente (Pacman) en un estado determinado (su ubicación o posición actual) dentro de un medio ambiente (el laberinto). La recompensa positiva que busca Pacman son los puntos por comer, mientras que la negativa será la de morir si se cruza con un fantasma, en base a la acción (desplazamiento a un nuevo estado) que realice (Merino, 2019).



**Figura 8.** Componentes del Aprendizaje por Refuerzo.

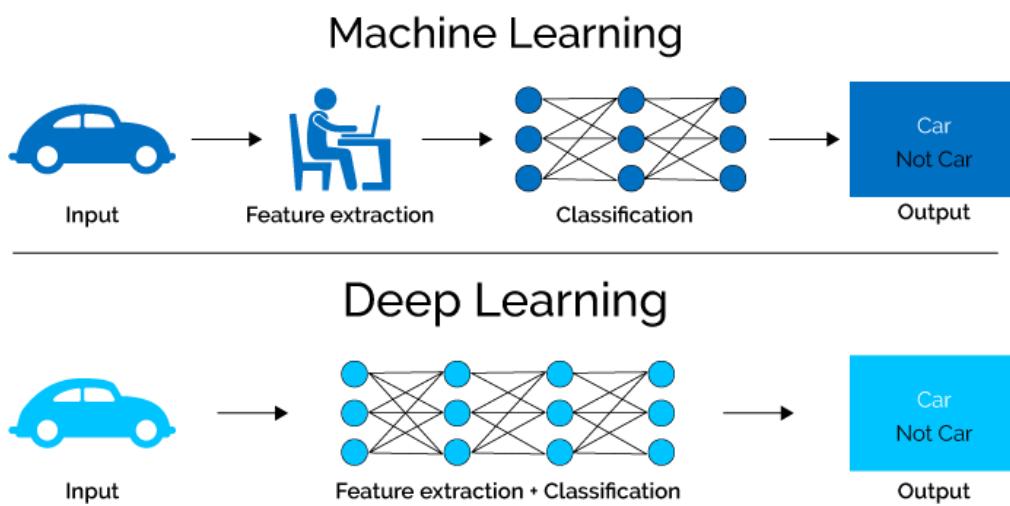
**Fuente:** (Sutton & Barto, 2018).

### 2.2.3. Aprendizaje Profundo

El Aprendizaje Profundo (*Deep Learning* por su nombre en inglés) es un tipo de Aprendizaje Automático que entrena a una computadora para que realice tareas como las realizadas por los seres humanos, desde la identificación de imágenes hasta realizar predicciones y reconocer el lenguaje humano. El Aprendizaje Profundo configura parámetros básicos acerca de los datos y entrena a la computadora para que aprenda por su cuenta reconociendo patrones mediante el uso de múltiples capas de procesamiento (SAS Institute). Se basa en teorías acerca de cómo funciona el cerebro humano (Banafa, 2019).

La principal diferencia con el Aprendizaje Automático es que el Aprendizaje Profundo se basa en la extracción de características y clasificación al mismo tiempo

luego de recibir una entrada, algo que en la primera técnica ocurre por separado, como se aprecia en la **Figura 9**.



**Figura 9.** Diferencia entre Aprendizaje Automático y Aprendizaje Profundo.

**Fuente:** (Cook, 2018).

Por un lado, mientras en el aprendizaje automático o de máquina, el ordenador extrae conocimiento a través de experiencia supervisada, en el aprendizaje profundo está menos sometido a supervisión. Mientras que el primer tipo de aprendizaje consume muchísimo tiempo y se basa en proponer abstracciones que permiten aprender al ordenador, en el segundo no consume demasiado tiempo y por el contrario de su par, crea redes neuronales a gran escala que permiten que el ordenador aprenda y piense por sí mismo sin necesidad directa de intervención humana. Actualmente, el aprendizaje profundo se usa para crear softwares capaces de determinar emociones o eventos descritos en textos, reconocimiento de objetos en fotografías y realizar predicciones acerca del posible comportamiento futuro de las personas. Empresas como Google (proyecto Google Brain) o Facebook (Unidad de investigación en IA) han puesto en marcha proyectos basados en esta rama para potenciar y mejorar sus algoritmos con el fin de ofrecer una mejor experiencia de sus servicios a sus clientes (Banafa, 2019).

#### 2.2.4. Modelo Predictivo

Son modelos de datos estadísticos utilizados para predecir el comportamiento futuro. En estos, se recopilan datos históricos y actuales, se formula un modelo estadístico, se realizan predicciones y el modelo se valida a medida que se dispone de

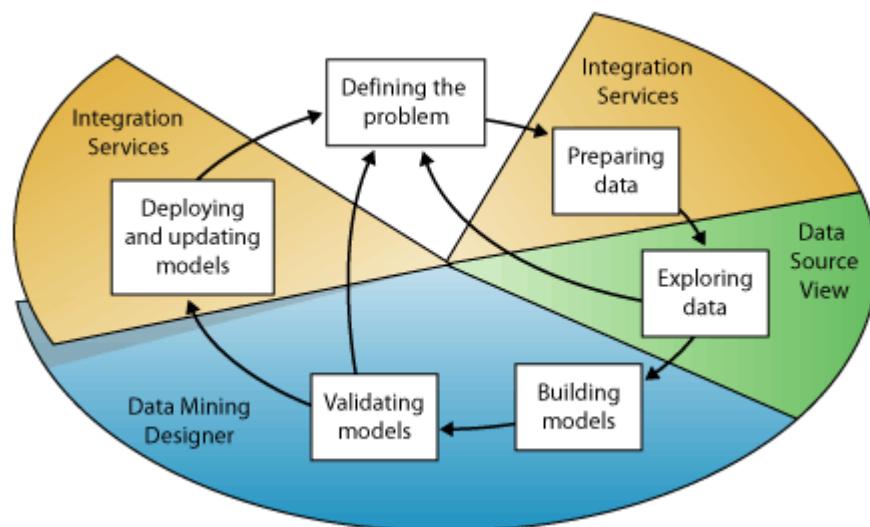
datos adicionales. Los modelos predictivos analizan el rendimiento pasado para evaluar la probabilidad de que un cliente muestre un comportamiento específico en el futuro. En esta categoría también abarca la búsqueda de patrones ocultos (Gartner, s.f.).

### 2.2.5. Minería de Datos

La Minería de Datos es un campo de la estadística y las ciencias de la computación referido al proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos (Maimon & Rokach, 2010). Normalmente, estos patrones no pueden detectarse mediante la exploración tradicional de datos porque sus relaciones son demasiado complejas o por su gran volumen. Para ello, utiliza métodos de Inteligencia Artificial, Aprendizaje Automático, estadística y sistemas de bases de datos. Estos patrones son recopilados y definidos como un modelo de minería de datos, los cuales pueden aplicarse en los siguientes escenarios (Microsoft, 2019):

- ✓ Previsión.
- ✓ Riesgo y probabilidad.
- ✓ Recomendaciones.
- ✓ Buscar secuencias.
- ✓ Agrupación.

La generación de un modelo de minería de datos forma de un macro-proceso descrita en los siguientes seis pasos representados en la **Figura 10** (Microsoft, 2019):



**Figura 10.** Diagrama de los seis pasos básicos.

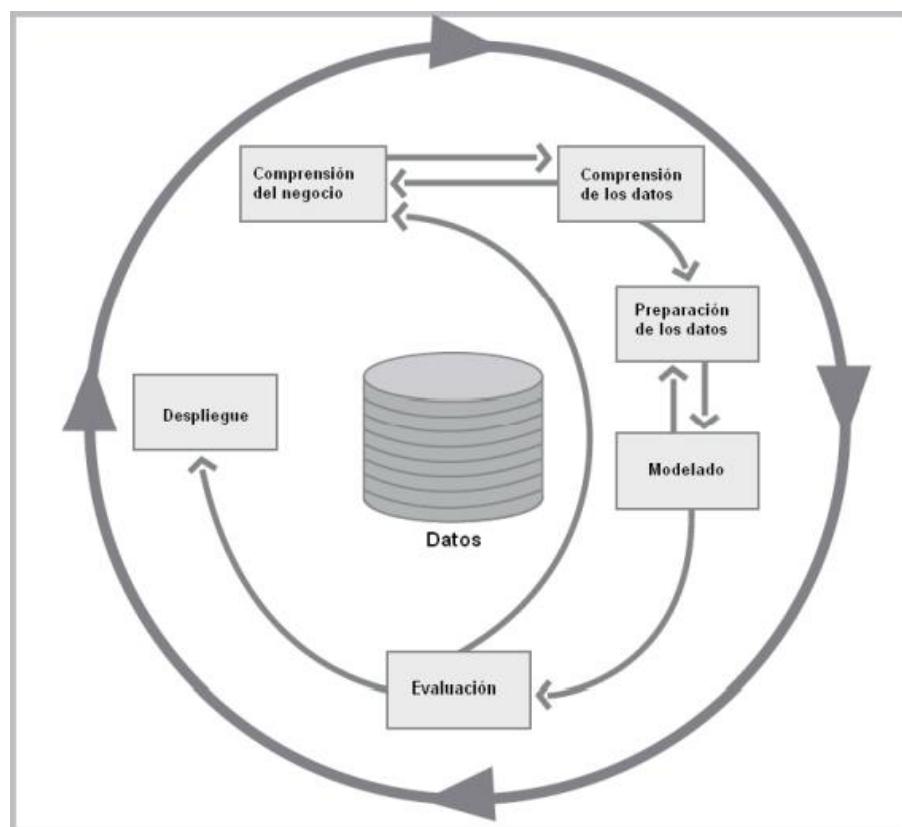
Fuente: (Microsoft, 2019).

### 2.2.6. Metodologías de Minería de Datos

Dentro de los sistemas de analítica de negocio, Big Data y Minería de Datos, las tres metodologías más usadas se encuentran CRISP-DM, SEMMA y KDD (Braulio Gil & Curto Díaz, 2015).

- ✓ **CRISP-DM** (Cross Industry Standard Process for Data Mining):

Esta metodología presenta seis fases representadas en la **Figura 11** a continuación.



**Figura 11.** Fases de la metodología CRISP-DM.

**Fuente:** (Braulio Gil & Curto Díaz, 2015).

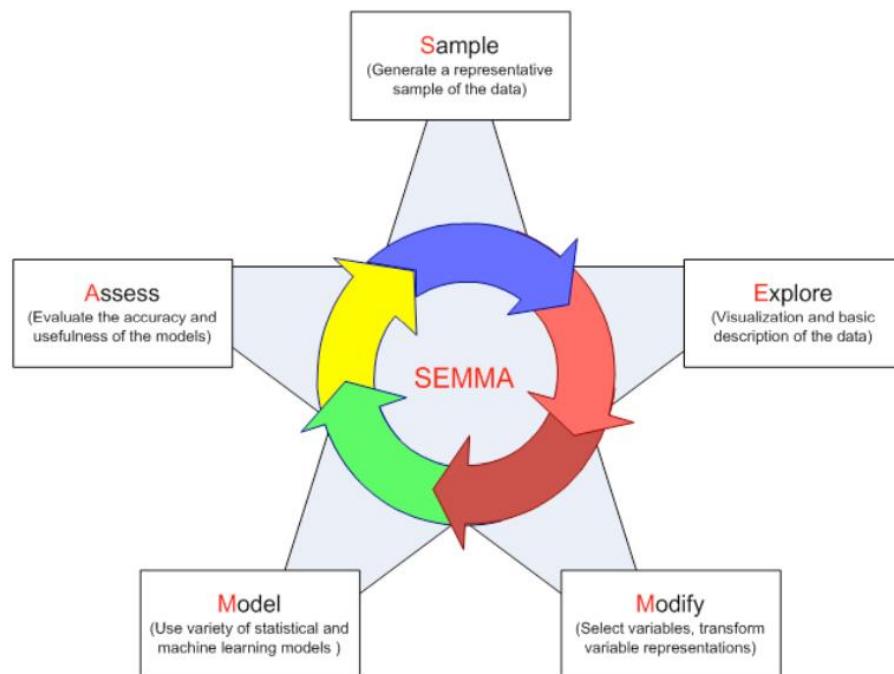
- En la comprensión del negocio se determinan los objetivos y requerimientos desde el lado del negocio, así como generar plan del proyecto.
- En la comprensión de los datos se logra entender el significado de las variables existentes, así como el entendimiento de los datos desde su recopilación hasta su verificación de calidad.
- En la preparación de los datos se prepara el conjunto de datos adecuado que servirán para la construcción del modelo. Por ello, la calidad de los

datos es un factor relevante y ello requiere la exclusión de redundancia y valores que no ayuden a establecer buena comprensión y resultados más adelante. A esto se le conoce como limpieza de datos.

- En el modelado se aplican técnicas de minería de datos en el conjunto de datos creado en el paso anterior. Para ello, se evalúan entre varias la que mejor performance desempeñe y luego se construye el o los modelos que busquen determinar un objetivo.
- En la evaluación se evalúan los posibles modelos del paso anterior a partir del nivel de importancia de acuerdo a las necesidades del negocio y performance que estos cuentan.
- El despliegue, finalmente, utiliza el modelo final creado para determinar los objetivos que se buscan cumplir en los requerimientos y ayudar en la toma de decisiones.

✓ **SEMMA** (Sample – Explore – Modify – Model – Assess):

Esta metodología cuenta con cinco fases como se aprecia en la **Figura 12**. A diferencia de la anterior, esta metodología se enfoca más en el modelado.



**Figura 12.** Fases de la metodología SEMMA.

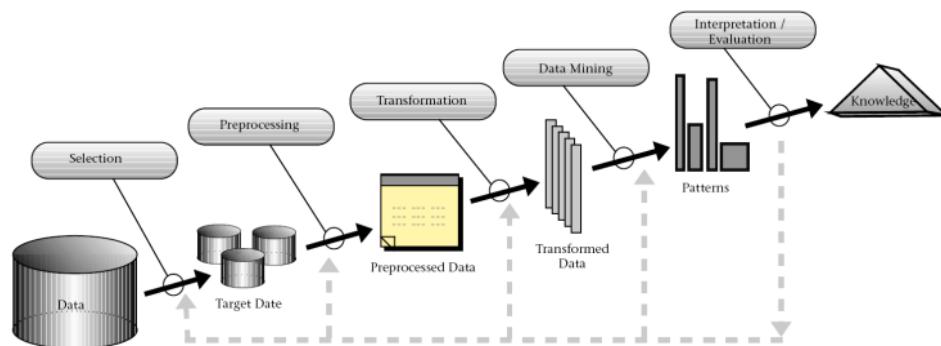
**Fuente:** (Braulio Gil & Curto Díaz, 2015).

- En la Muestra (Sample) se crea una muestra significativa.

- En la Exploración (Explore) se comprenden los datos con el fin de encontrar relaciones entre variables y anomalías.
- En la Modificación (Modify) se transforman las variables para las necesidades del modelo.
- En la Modelización (Model) se aplican uno o varios modelos sobre el conjunto de datos para buscar resultados.
- En el Asesoramiento (Assessment) se evalúan los resultados obtenidos del modelo.

✓ **KDD (Knowledge Discovery and Data Mining):**

Esta metodología se refiere al proceso de encontrar conocimiento alguno en el dato y, a diferencia de sus predecesores, se enfoca en crear aplicaciones de minería de datos. Consta de cinco fases más 1 previa y 1 posterior basadas en la generación de conocimiento como se muestra en la **Figura 13**.



**Figura 13.** Fases de la metodología KDD.

**Fuente:** (Braulio Gil & Curto Díaz, 2015)

- En la fase Pre KDD se comprende el dominio del negocio, así como también se identifican las necesidades del cliente.
- En la selección, primero se identifica el conjunto de datos a usar y luego se seleccionan la muestra y las variables para la exploración.
- En el pre-procesamiento, se realiza la limpieza de datos y se elimina el ruido, así como los valores atípicos.
- En la transformación se implementan métodos de reducción de dimensiones para reducir el número de variables efectivas.

- En la Minería de datos, se elige el tipo de tarea de minería de datos (clasificación, regresión, agrupamiento, entre otros) así como el algoritmo, los métodos, los modelos y parámetros apropiados.
- En la interpretación y evaluación se analizan los resultados dados.
- En la fase Post KDD finalmente se consolida el conocimiento adquirido.

Luego de presentar las tres metodologías más usadas, la pregunta dada es ¿cuál de los tres representa la mejor opción para usar?

Las tres metodologías tienen distinto número de pasos, así como distintos enfoques, tal cual se observa en el siguiente resumen de la **Tabla 2**.

<b>Modelos de Procesos de Minería de Datos</b>	<b>KDD</b>	<b>CRISP-DM</b>	<b>SEMMA</b>
Número de pasos	9	6	5
Nombre de los pasos	Desarrollo y entendimiento de la aplicación	Entendimiento del negocio	-
	Creación de un conjunto de datos de destino	Entendimiento de los datos	Muestreo
	Limpieza de datos y pre-procesamiento		Exploración
	Transformación de datos	Preparación de los datos	Modificación
	Elección de la tarea adecuada de Minería de datos	Modelamiento	
	Elección del algoritmo adecuado de Minería de datos		Modelo
	Implementación del algoritmo de Minería de datos		
	Interpretación de patrones minados	Evaluación	Evaluación
	Uso de conocimiento descubierto	Despliegue	-

**Tabla 2.** Cuadro comparativo entre características de las tres metodologías.

**Fuente:** (Shafique & Qaiser, 2014)

Sin embargo, la elección depende de los involucrados que finalmente usarán el modelo en el negocio. La mayoría de investigadores siguen la metodología KDD debido a que es más completo y su exactitud. Para aquellos objetivos enfocados más en la compañía como la integración usada por SAS Enterprise Miner con su software se utilizan SEMMA y CRISP-DM. Esta última resulta ser más completa de acuerdo a los estudios.

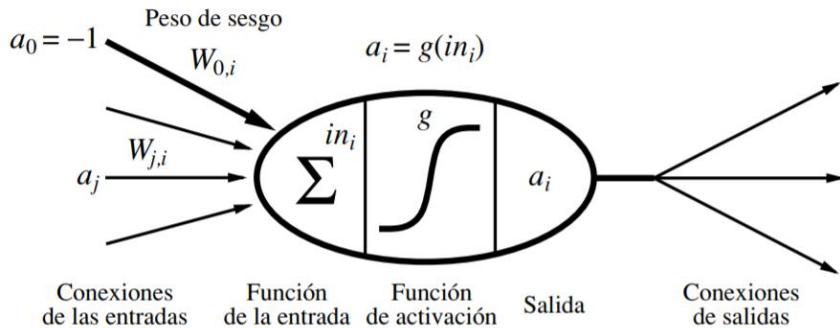
### 2.2.7. Técnicas de Minería de Datos

Existe una gran variedad de técnicas para la Minería de Datos. Las más importantes y utilizadas en los antecedentes de la investigación se mencionan a continuación (Microsoft, 2018).

- ✓ **Redes Neuronales Artificiales (RNA):** Es un sistema de computación que consiste en un número de elementos o nodos simples, pero altamente interconectados, llamados “neuronas”, que se organizan en capas que procesan información utilizando respuestas de estado dinámico a entradas externas (Inzaugarat, 2018).

Este sistema de programas y estructura de datos se aproxima al funcionamiento del cerebro humano. Una red neuronal implica tener un gran número de procesadores funcionando en paralelo, teniendo cada uno de ellos su propia esfera de conocimiento y acceso a datos en su memoria local. Normalmente, una se alimenta con grandes cantidades de datos y un conjunto dado de reglas acerca de las relaciones. Luego, un programa puede indicar a la red cómo debe comportarse en respuesta a un estímulo externo o si puede iniciar la actividad por sí misma (Banafa, 2019).

Para entender mejor cómo funciona una red neuronal, hay que describir qué es una neurona. Una neurona es una célula del cerebro cuya función principal es la recogida, procesamiento y emisión de señales eléctricas. Debido a que se piensa que la capacidad de procesamiento de información del cerebro proviene de redes de este tipo de neuronas, los primeros trabajos en Inteligencia Artificial se basaron en crear redes neuronales artificiales para emular este comportamiento, en 1943 con un modelo matemático, mostrado en la **Figura 14**, por los ya mencionados anteriormente McCulloch y Pitts. Estos y posteriores trabajos potenciaron lo que hoy en día se conoce como el campo de la neurociencia computacional (Russell & Norvig, Inteligencia Artificial: Un Enfoque Moderno, 2004). Años más tarde, en 1958, se desarrolló el concepto del perceptrón por Rosenblatt, el cual tenía la capacidad de aprender y reconocer patrones sencillos, formado por entradas, neurona, función de adaptación (sigmoidal, tangencial, en escalón, etc.) y salida.



**Figura 14.** Modelo para representar una neurona propuesto por McCulloch y Pitts (1943).

**Fuente:** (Russell & Norvig, Inteligencia Artificial: Un Enfoque Moderno, 2004).

La última figura descrita muestra, además de los pesos, funciones de activación tanto para la entrada ( $a_j$ ) como para la salida ( $a_i$ ). Pero, ¿qué son estas funciones y para qué sirven?

Para comenzar, las redes neuronales están compuestas de nodos (la elipse) conectados a través de conexiones dirigidas (las flechas). Una conexión del nodo  $j$  a la unidad  $i$  sirve para propagar la activación  $a_j$  de  $j$  a  $i$ . Asimismo, cada conexión tiene un peso numérico  $W_{j,i}$  que determina la fuerza y el signo de la conexión. Para calcular cada nodo  $i$ , se realiza una suma ponderada de sus entradas (producto entre pesos y nodos de entrada  $j$ ), y se le añade el sesgo (*bias*)  $\theta_i$  (aumenta/disminuye el valor de la combinación lineal de las entradas):

$$in_i = \sum_{j=0}^n W_{j,i} * a_j + \theta_i$$

**Ecuación 4.** Fórmula del cálculo del valor de un nodo  $i$ .

**Fuente:** (Russell & Norvig, Inteligencia Artificial: Un Enfoque Moderno, 2004).

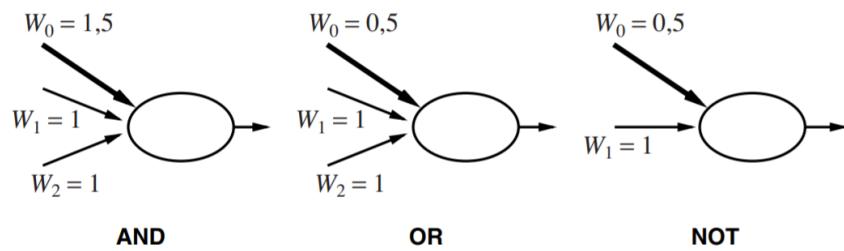
Posteriormente, se efectúa una función de activación  $g$  a esta suma para producir la salida:

$$a_i = g(in_i) = g\left(\sum_{j=0}^n W_{j,i} * a_j + \theta_i\right)$$

**Ecuación 5.** Fórmula de una función de activación  $g$  para la salida del nodo.

**Fuente:** (Russell & Norvig, Inteligencia Artificial: Un Enfoque Moderno, 2004).

Entonces, aquí se explica los dos objetivos de una función de activación. En primer lugar, se desea que el nodo esté “activo” (cercano a +1) cuando las entradas correctas sean dadas, e “inactiva” (cercano a 0) cuando las entradas erróneas sean proporcionadas. En segundo lugar, la activación tiene que ser no lineal porque, de lo contrario, la red neuronal colapsaría en su totalidad con una función lineal sencilla, como se aprecia en el ejemplo de la **Figura 15**.



**Figura 15.** Nodos con funciones de activación umbral en forma de puertas lógicas.

**Fuente:** (Russell & Norvig, Inteligencia Artificial: Un Enfoque Moderno, 2004).

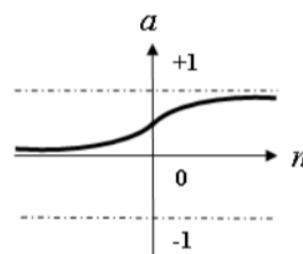
Entre las funciones de activación que más destacan son las siguientes:

- **Función sigmoide o logística:** Toma los valores de entrada que oscilan entre infinito negativo y positivo, y restringe los valores de salida al rango entre 0 y 1. Frecuentemente es usada en Redes Multicapa (MLP) entrenadas con el algoritmo de propagación inversa. Se representa como en la **Figura 16** y su fórmula para calcular su nuevo valor es:

$$a = \text{Logsig}(n) = \frac{1}{1 + e^{-n}}$$

**Ecuación 6.** Fórmula de la función de activación sigmoide.

**Fuente:** (Dorofki, Elshafie, Jaafar, Karim, & Mastura, 2012).



**Figura 16.** Función de activación sigmoide.

**Fuente:** (Dorofki, Elshafie, Jaafar, Karim, & Mastura, 2012).

Un dato curioso de esta función relacionado con la regresión logística es que el nombre de esta última no deriva de una regresión. Por el contrario, se debe a que, al principio de la neurona, se realiza una combinación lineal muy parecida a una regresión lineal y después se aplica la función logística o sigmoide. De ahí el origen del nombre (IArtificial.net, 2019).

- ❖ **Regresión Logística:** Como se mencionó antes, es similar a un modelo de regresión lineal, pero está adaptado para modelos en los que la variable dependiente es dicotómica, es decir, presenta solo dos posibles valores. Resulta muy útil para los casos en los que se desea predecir la presencia o ausencia de una característica o resultado según los valores de un conjunto de predictores (International Business Machines Corporation (IBM)). Su función de coste que se optimiza con gradiente descendiente se representa mediante la siguiente fórmula:

$$J = -\frac{1}{m} \sum_i^m y_i * \log(\hat{p}_i) + (1 - y_i) * \log(1 - \hat{p}_i)$$

**Ecuación 7.** Fórmula de función de coste de una regresión logística.

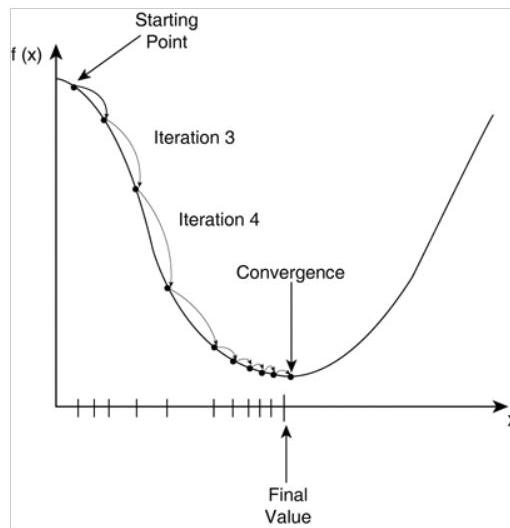
**Fuente:** (IArtificial.net, 2019).

Donde la primera parte de la ecuación está conformada por el logaritmo de la probabilidad de éxito y la segunda, por la de fracaso.

- ❖ **Gradiente descendiente:** Es un método de optimización numérica para estimar los mejores coeficientes, fundamental en Deep Learning para entrenar redes neuronales y en muchos casos, para la regresión logística, siendo mejor que el método de mínimos cuadrados (IArtificial.net). A través de una función E(W), proporciona el error que comete la red en función del conjunto de pesos sinápticos W. El objetivo del aprendizaje será encontrar la configuración de pesos que corresponda al mínimo global de la función de error o coste (Bertona, 2005).

En general, la función de error es una función no lineal, por lo que el algoritmo realiza una búsqueda a través del espacio de parámetros que, se aproxime de forma iterada a un error mínimo de la red para

los parámetros adecuados, como se aprecia en la **Figura 17** (Sancho Caparrini, Entrenamiento de Redes Neuronales: mejorando el Gradiente Descendiente, 2017).



**Figura 17.** Ilustración del algoritmo gradiente descendiente.

**Fuente:** (Sancho Caparrini, Entrenamiento de Redes Neuronales: mejorando el Gradiente Descendiente, 2017).

El Descenso del Gradiente, como también se le conoce, es el algoritmo de entrenamiento más simple y también el más extendido y conocido. Solo hace uso del vector gradiente, y por ello se dice que es un método de primer orden (Sancho Caparrini, Entrenamiento de Redes Neuronales: mejorando el Gradiente Descendiente, 2017). Un gradiente es la generalización de la derivada. Matemática, la derivada de una función mide la rapidez con la que cambia el valor de esta, según varíe el valor de su variable independiente. La gradiente se calcula con derivadas parciales, por lo que al actualizar los coeficientes W para un tiempo  $t$ , se usa la regla (IArtificial.net):

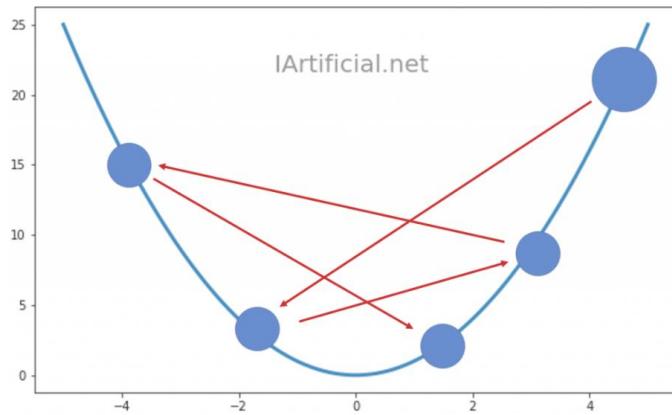
$$W_{(t+1)} = W_{(t)} - \alpha \left( \frac{\partial MSE}{\partial W} \right)$$

**Ecuación 8.** Actualización de pesos W mediante gradiente descendiente.

**Fuente:** (IArtificial.net).

Donde  $\alpha$  es el “ratio de aprendizaje”, el cual controla el tamaño de la actualización, si este es demasiado grande será más difícil encontrar

los coeficientes que minimicen la función de coste o error; la actualización de  $W$  es proporcional al gradiente; y se usa la resta para ir en dirección opuesta al gradiente como en la **Figura 18**.



**Figura 18.** Actualización de pesos  $W$  con el algoritmo.

**Fuente:** (IArtificial.net).

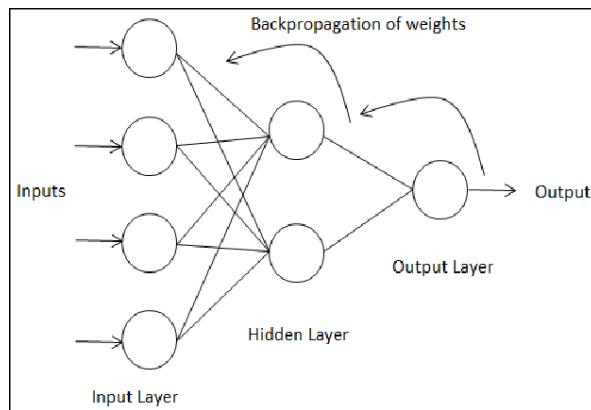
❖ **Propagación hacia atrás:** También conocido en inglés como *Backpropagation*, es un método que consta de dos fases: en la primera se aplica un patrón, el cual se propaga por las distintas capas que componen la red hasta producir la salida de la misma. Luego, esta se compara con la salida deseada y se calcula el error cometido por cada neurona de salida. Estos errores se transmiten hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de las capas intermedias [Fritsch, 1996] (Bertona, 2005). La actualización iterativa de los pesos que el algoritmo propone es mediante la siguiente fórmula:

$$W_{ji}(t+1) = W_{ji}(t) + [\alpha \delta_{pj} y_{pj} + \beta \Delta W_{ji}(t)]$$

**Ecuación 9.** Fórmula del algoritmo de propagación hacia atrás.

**Fuente:** (Bertona, 2005).

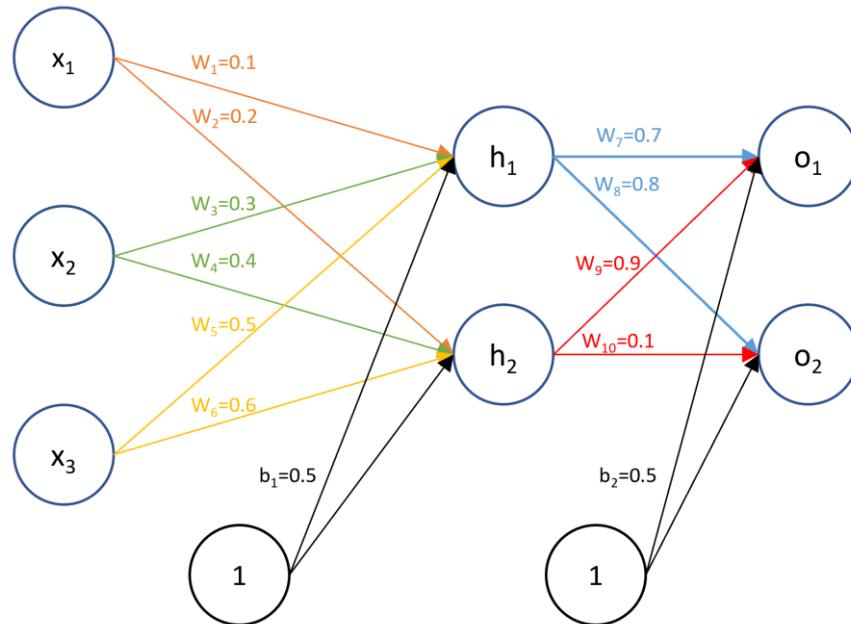
$$\text{siendo } \delta_{pj} = \begin{cases} (d_{pj} - y_{pj})f'_j(h_j) & \text{si } j \text{ es una neurona de salida} \\ \left( \sum_k \delta_{pk} w_{kj} \right) f'_j(h_j) & \text{si } j \text{ es una neurona oculta} \end{cases}$$



**Figura 19.** Capa oculta simple MLP con propagación hacia atrás.

**Fuente:** (Mohmad Hassim & Ghazali, 2012)

Para entender mejor la teoría y la fórmula de actualización de pesos, se seguirá el siguiente ejemplo del conjunto de redes de la **Figura 20**.



**Figura 20.** Redes neuronales de ejemplo.

**Fuente:** (A Not So Random Walk, 2019)

Se tiene una red neuronal con tres nodos de entrada ( $x_1 = 1, x_2 = 4$  y  $x_3 = 5$ ) con dos pesos respectivos cada una ( $W_1 = 0.1$  y  $W_2 = 0.2$  para  $x_1$ ;  $W_3 = 0.3$  y  $W_4 = 0.4$  para  $x_2$ ;  $W_5 = 0.5$  y  $W_6 = 0.6$  para  $x_3$ ), dos capas ocultas ( $h_1$  y  $h_2$ ) con dos peso cada una ( $W_7 = 0.7$  y  $W_8 = 0.8$  para  $h_1$ ;  $W_9 = 0.9$  y  $W_{10} = 0.1$  para  $h_2$ ) y dos nodos de salida ( $o_1$  y  $o_2$ ). El proceso normal para calcular el valor del nodo final se da, tanto con los nodos de entrada y los de capa oculta, mediante la sumatoria de

producto de cada peso con su valor, es decir, mediante la fórmula de las RNA  $in_i = \sum_{j=0}^n W_{j,i} * a_j + b_{j,i}$ , al mismo tiempo que devuelve un valor del error cometido. Este último se calcula mediante la siguiente ecuación:

$$E_k = (T_k - O_k) * O_k * (1 - O_k)$$

**Ecuación 10.** Cálculo del error cometido en una red neuronal.

**Fuente:** (Viera Balanta, 2013).

Donde  $T_k$  es la salida correcta de cada nodo de salida, y  $O_k$  es la salida actual que cada uno genera. Con estos errores calculados, se retrocede hacia la capa oculta y se procede a calcular los nuevos pesos para sus nodos. La fórmula del cálculo de los mismos es:

$$W_{jk} = W_{jk} + L * E_k * O_j$$

**Ecuación 11.** Actualización de pesos mediante propagación hacia atrás.

**Fuente:** (Viera Balanta, 2013).

Donde  $W_{jk}$  representa el peso para cada nodo de la capa oculta, es decir,  $W_7, W_8, W_9$  y  $W_{10}$ , los mismos que serán actualizados, L es el porcentaje de aprendizaje y  $O_j$  son los valores de estos dos nodos que entrarán a las salidas. Estos nuevos pesos permitirán redefinir los errores de ambos nodos, con una pequeña diferencia en su cálculo:

$$E_j = O_j * (1 - O_j) * \sum E_k * W_{jk}$$

**Ecuación 12.** Cálculo de errores de nodos usando pesos actualizados.

**Fuente:** (Viera Balanta, 2013).

El error de cada nodo de la capa oculta se obtiene multiplicando su valor por su complemento por la sumatoria del producto de sus pesos y los errores de los nodos de salida. Por ejemplo, para  $h_1$  sería  $E_{h1} = h_1 * (1 - h_1) * (0.7 * o_1 + 0.8 * o_2)$ .

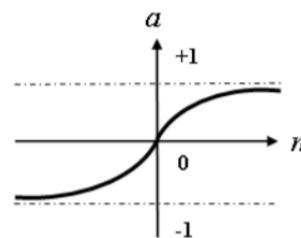
Finalmente, se retrocede hacia los nodos de entrada y se repite el mismo proceso para la actualización de sus pesos y errores.

- **Función tangente hiperbólica:** Esta función está relacionada con una sigmoide bipolar. Sin embargo, sus salidas estarán en el rango de -1 y +1. Para redes neuronales, donde la velocidad es más importante que la forma de la función misma, es recomendable usar esta. Se representa como en la **Figura 21** y su fórmula para calcular su nuevo valor es:

$$a = \text{Tansig}(n) = \frac{2}{1 + e^{-2n}} - 1$$

**Ecuación 13.** Fórmula de la función de activación tangente hiperbólica.

**Fuente:** (Dorofki, Elshafie, Jaafar, Karim, & Mastura, 2012).



**Figura 21.** Función de activación tangente hiperbólica.

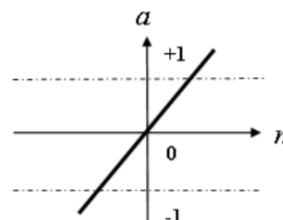
**Fuente:** (Dorofki, Elshafie, Jaafar, Karim, & Mastura, 2012).

- **Función puramente lineal (purelin):** Esta función se caracteriza porque su salida es igual a su entrada debido a su linealidad. Normalmente se usa para obtener los mismos valores de la entrada. Se representa como en la **Figura 22** y su fórmula para calcular su nuevo valor es:

$$a = n$$

**Ecuación 14.** Fórmula de la función de activación puramente lineal.

**Fuente:** (Dorofki, Elshafie, Jaafar, Karim, & Mastura, 2012).



**Figura 22.** Función de activación puramente lineal.

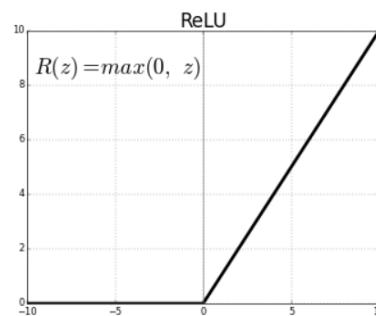
**Fuente:** (Dorofki, Elshafie, Jaafar, Karim, & Mastura, 2012).

➤ **Función Unidad Lineal Rectificada (ReLU):** Esta función se caracteriza por, además de conservar los valores positivos, convertir los valores negativos de entrada en 0, esto con la finalidad de no considerarlos en la siguiente capa de convolución como en el caso de procesamiento de imágenes (SitioBigData.com, 2019). Si bien tiene un buen desempeño en redes convolucionales y es muy usada para el procesamiento de imágenes, al no estar acotada pueden morirse demasiadas neuronas (Calvo, Función de activación - Redes neuronales, 2018). Se representa como en la **Figura 23** y su fórmula para calcular su nuevo valor es:

$$f(x) = \max(0, x) = \begin{cases} 0 & \text{para } x < 0 \\ x & \text{para } x \geq 0 \end{cases}$$

**Ecuación 15.** Fórmula de la función de activación ReLU.

**Fuente:** (Calvo, Función de activación - Redes neuronales, 2018).



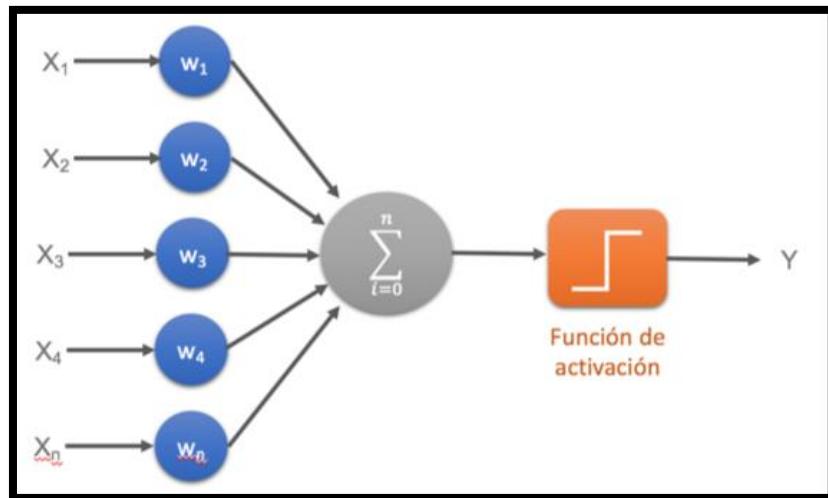
**Figura 23.** Función de activación Unidad Lineal Rectificada.

**Fuente:** (Machine Learning for Artists)

Además de existir distintas funciones de activación, las redes neuronales artificiales se clasifican según la topología de red, siendo algunas de las más importantes (Calvo, Clasificación de redes neuronales artificiales, 2017):

- ❖ **Red Neuronal Monocapa – Perceptrón simple:** Es la red neuronal más simple ya que está compuesta solamente de una capa de neuronas que componen varios nodos de entrada para proyectar una capa de neuronas de salida, como se aprecia en la **Figura 24**. Esta última capa se calcula usando la misma **Ecuación 4** que implica la suma de productos de cada uno de los pesos de los nodos de entrada con sus instancias, añadiéndole

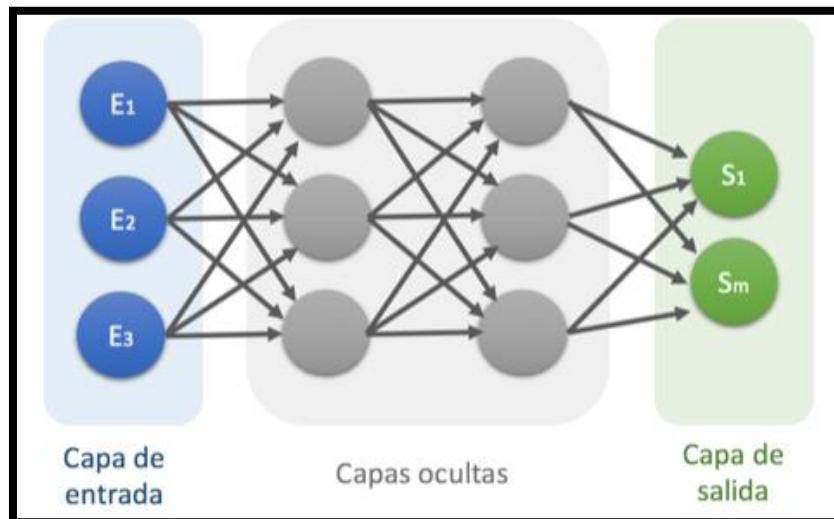
finalmente el sesgo, aquel que controla la predisposición de la neurona a disparar un 1 o 0 independientemente de los pesos, para que el valor resultante se le aplique la función de activación que ayudarán a modelar funciones curvas o no triviales (Machine Learning for Artists).



**Figura 24.** Ejemplo de perceptrón simple.

**Fuente:** (Calvo, Clasificación de redes neuronales artificiales, 2017).

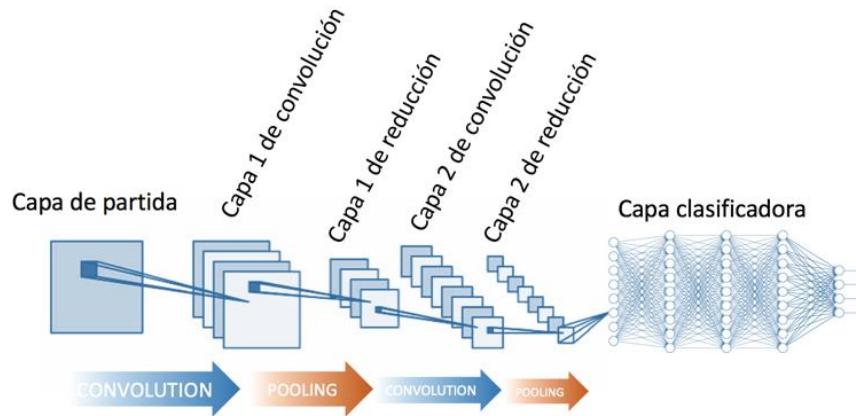
- ❖ **Red Neuronal Multicapa – Perceptrón multicapa:** Con arquitectura similar al perceptrón simple, con el añadido de contener capas intermedias entre la capa de neuronas de entrada y la de salida, conocidas como capas ocultas, como en el ejemplo de la **Figura 25**.



**Figura 25.** Ejemplo de perceptrón multicapa.

**Fuente:** (Calvo, Clasificación de redes neuronales artificiales, 2017).

- ❖ **Redes Neuronales Convolucionales (CNN):** También conocidas por su nombre en inglés *Convolutional Neural Networks*, se diferencia del perceptrón multicapa en que cada neurona no necesita estar unida con todas las que le siguen, sino más bien solo con un subgrupo de estas con el fin de reducir la cantidad de neuronas necesarias para su funcionamiento, como se observa en la **Figura 26** (Calvo, Clasificación de redes neuronales artificiales, 2017).

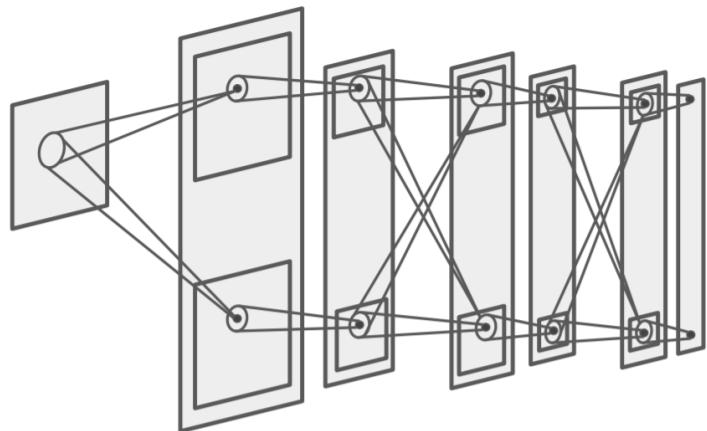


**Figura 26.** Ejemplo de red neuronal convolucional.

**Fuente:** (Calvo, Clasificación de redes neuronales artificiales, 2017).

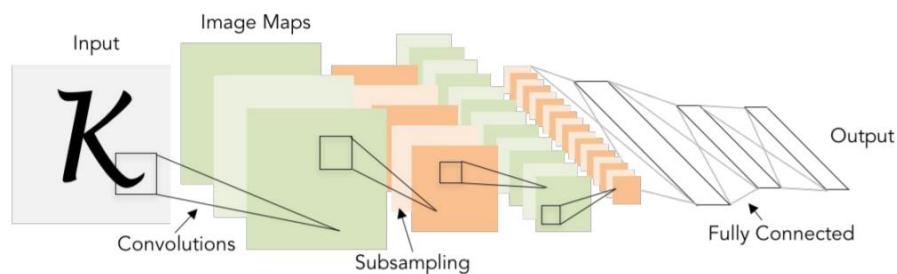
Hoy en día, las redes neuronales convolucionales tienen múltiples usos desde que la idea fue concebida. Algunos de los problemas en las que pueden ser usados son de clasificación de objetos, recuperación de imágenes, detección y segmentación de objetos, distorsión y filtros de imágenes, por citar los ejemplos más comunes. El modelo de CNN más conocido es “AlexNet” (2012) por ser uno de los pioneros en clasificar imágenes (Li, Johnson, & Yeung, 2019).

Estas redes tienen su origen en el Neocognitron introducido por Fukushima en 1980 como modelo de red neuronal para el mecanismo de reconocimiento de patrón visual sin la enseñanza de un “profesor” (ver **Figura 27**), mismo que en el año 1998 sería mejorado por LeCun, Bottou, Bengio y Haffner al agregar un método de aprendizaje de gradiente aplicado al reconocimiento de documento basado en la propagación hacia atrás (ver **Figura 28**) (Li, Johnson, & Yeung, 2019).



**Figura 27.** Modelo Necognitron de Fukushima (1980).

**Fuente:** (Li, Johnson, & Yeung, 2019).



**Figura 28.** Modelo LeNet-5 de LeCun (1998).

**Fuente:** (Li, Johnson, & Yeung, 2019).

Estos modelos se inspiraron en el estudio de la información visual en la corteza donde se ubican hasta 5 áreas. La primera, V1, contiene la información visual donde sus neuronas se ocupan de características visuales de bajo nivel, alimentando así a otras áreas adyacentes. Cada una de ellas se encarga de aspectos más específicos y detallados de la información obtenida. La idea de su implementación es la de solucionar el problema que surgen al escalar imágenes de mucha definición por las redes neuronales ordinarias. Por ello, este tipo de redes trabajan modelando de forma consecutiva piezas pequeñas de información para luego combinarlas en sus capas más profundas (López Briega, 2016).

Su nombre deriva del concepto convolución. La convolución es un término en las matemáticas usado como operador matemático que

convierte dos funciones  $f$  y  $g$  en una tercera función en donde la primera se superpone a una versión invertida y trasladada de la segunda, así como para denotar la distribución de la función de probabilidad de la suma de dos variables independientes aleatorias. Esta se da por la siguiente fórmula (Figueroa M.):

$$(f * g)(t) = \int_0^t f(t - \tau)g(\tau)d\tau$$

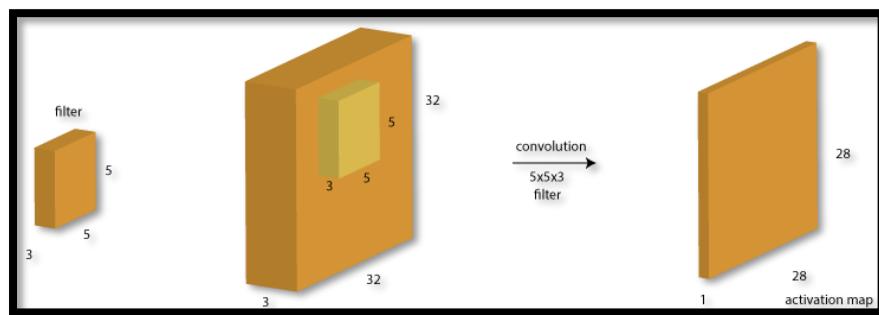
**Ecuación 16.** Fórmula matemática de la convolución.

**Fuente:** (Figueroa M.).

Donde el rango puede variar entre un conjunto finito (como en la fórmula desde 0 hasta un valor t) o uno infinito.

La estructura de las Redes Neuronales Convolucionales se constituye en tres tipos de capas (López Briega, 2016):

- **Capa convolucional (*Convolutional Layer*):** Es la capa que hace distinta a esta red de otros tipos de redes neuronales artificiales. Se aplica la operación de la convolución, que recibe como entrada (*input* en inglés) a la imagen para luego aplicarle un filtro (*kernel* en inglés), devolviendo un mapa de las características de la imagen original, logrando así reducir el tamaño de los parámetros, como se observa en la **Figura 29**.

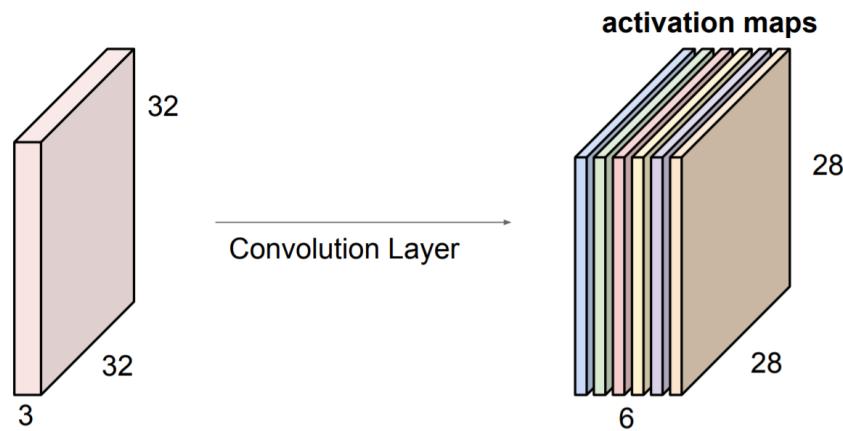


**Figura 29.** Ejecución de la convolución en una entrada.

**Fuente:** (López Briega, 2016).

Por ejemplo, en la anterior figura se tiene una imagen de entrada con dimensiones de 32 de alto, 32 de ancho y 3 de

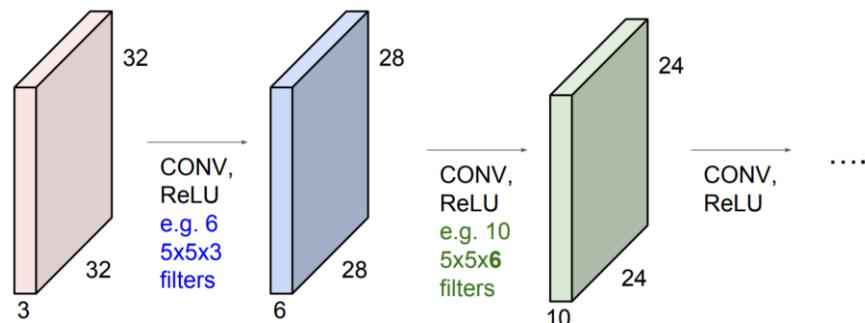
profundidad ( $32 \times 32 \times 3$ ). A ella se le aplica un filtro de dimensiones ( $5 \times 5 \times 3$ ) que recorrerá toda la imagen para extraer características de cada pixel. Tanto la profundidad de la entrada como del filtro siempre son iguales. El resultado de tomar un producto escalar entre el filtro y un pequeño fragmento de  $5 \times 5 \times 3$  de la imagen es un número, generando así un mapa de activación de nuevas dimensiones ( $28 \times 28 \times 1$ ). Por cada  $n$  filtros aplicados a la entrada se generan  $n$  de estos mapas. Al final, la cantidad de mapas de activación determinará una nueva imagen de  $n$  de profundidad, como en la **Figura 30**.



**Figura 30.** Generación de una nueva imagen a partir de filtros.

**Fuente:** (Li, Johnson, & Yeung, 2019).

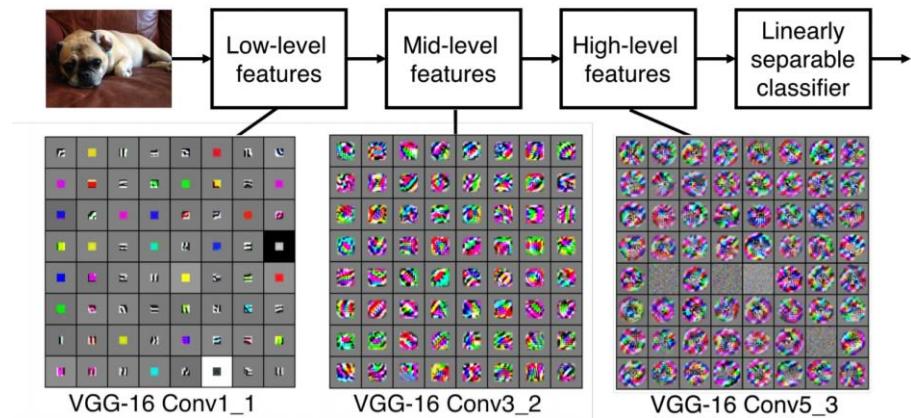
Asimismo, cada vez que se aplica una convolución a una imagen, se aplicará una función de activación como en la secuencia de la **Figura 31**.



**Figura 31.** Secuencia de varias capas convolucionales.

**Fuente:** (Li, Johnson, & Yeung, 2019).

A nivel visual, en la **Figura 32** se aprecia un ejemplo de los resultados de aplicar varias convoluciones a una imagen.



**Figura 32.** Extracción de características a partir de convoluciones.

**Fuente:** (Li, Johnson, & Yeung, 2019).

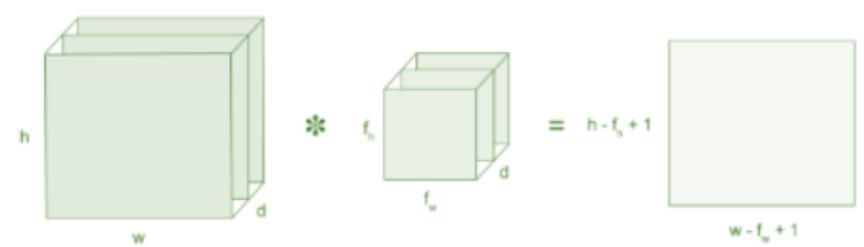
Finalmente, se calcula el volumen de la dimensión de la salida de la **Figura 33** mediante la siguiente ecuación:

- Se tiene una entrada de dimensiones ( $h \times w \times d$ ).
- Se tiene un filtro de dimensiones ( $f_h \times f_w \times d$ ).

$$\text{Volumen} = (h - f_h + 1) \times (w - f_w + 1) \times 1$$

**Ecuación 17.** Cálculo del volumen del mapa de activación.

Fuente: (Prabhu, 2018).



**Figura 33.** Ejemplo de matriz de imagen de entrada y un filtro.

**Fuente:** (Prabhu, 2018).

- **Capa de reducción (Pooling Layer):** Esta capa le sucede a la capa convolucional (luego de aplicar la función de activación). Sirve principalmente para reducir las dimensiones espaciales del volumen de la entrada (alto x ancho) para la siguiente capa convolucional. Sin embargo, no afecta la profundidad de la misma. Esta operación que

realiza se le conoce también como “reducción de muestreo” debido a que, si bien logra reducir las dimensiones para procesar mejor en la siguiente capa, también conlleva perder información. Por el contrario de lo que se piensa, además de reducir la sobrecarga del cálculo para las siguientes capas, el modelo se beneficia también disminuyendo el sobreajuste.

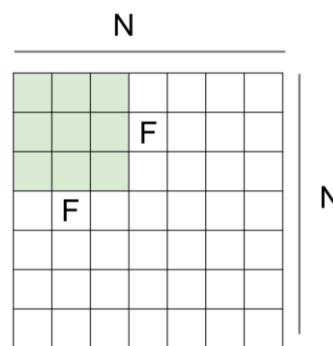
Para determinar las dimensiones de la nueva imagen generada (siempre que sea cuadrada, es decir, lados iguales como en la **Figura 34**) con esta capa, se aplica la siguiente fórmula:

$$\text{Tamaño de salida} = \frac{N - F}{Paso} + 1$$

**Ecuación 18.** Cálculo del tamaño de la imagen reducida.

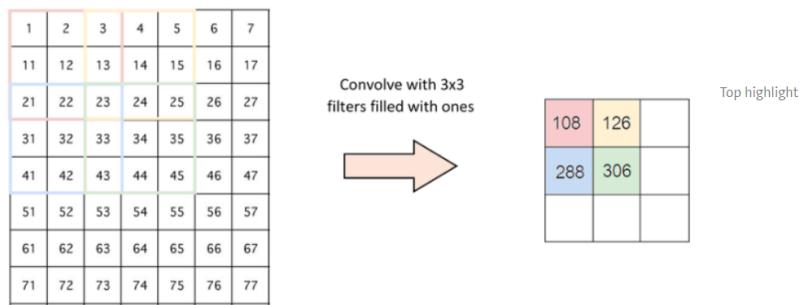
**Fuente:** (Li, Johnson, & Yeung, 2019).

Donde  $N$  es el tamaño del lado de la imagen de entrada,  $F$  es el tamaño del lado del filtro y  $Paso$  (*Stride* en inglés) es el número de desplazamiento de píxeles sobre la matriz de entrada. Por ejemplo, cuando el paso es 1, los filtros se mueven a 1 pixel por vez, cuando el paso es 2, se mueven a 2 píxeles (como en la **Figura 35**) y así sucesivamente (Prabhu, 2018).



**Figura 34.** Dimensiones de una entrada y un filtro.

**Fuente:** (Li, Johnson, & Yeung, 2019).



**Figura 35.** Paso de 2 píxeles por parte de un filtro.

**Fuente:** (Prabhu, 2018).

Si, por el contrario, se desea aplicar convolución a una imagen sin afectar sus dimensiones luego de pasar por la capa de reducción, se construye bordes de ceros de  $n$  píxeles. A este tamaño de borde se le llama Relleno (*pad* en inglés), por lo que el tamaño de la nueva salida se obtiene mediante la siguiente fórmula:

$$\text{Tamaño de salida} = \frac{N - F + 2 * \text{Relleno}}{\text{Pasos}} + 1$$

**Ecuación 19.** Cálculo del tamaño de la imagen reducida con bordes rellenos de ceros.

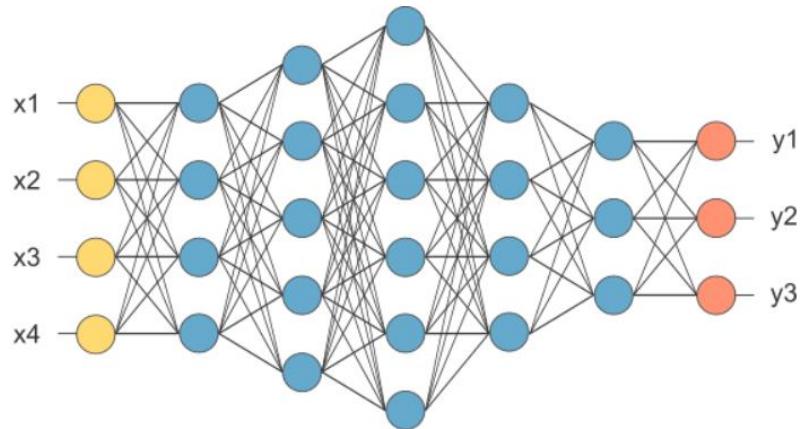
**Fuente:** (Li, Johnson, & Yeung, 2019).

Existen diferentes tipos de reducción (Prabhu, 2018):

- Max Pooling: Toma el elemento más grande dentro del mapa de características.
- Average Pooling: Toma el promedio de los elementos dentro del mapa de características.
- Sum Pooling: Toma la suma total de los elementos dentro del mapa de características.

➤ **Capa totalmente conectada (*Fully Connected Layer*):** Al final de las capas de convolución y reducción, se usan redes completamente conectadas a cada pixel considerando que cada uno como una neurona separada al igual que en una red neuronal regular (López Briega, 2016). En esta capa, se aplana la matriz de todas las

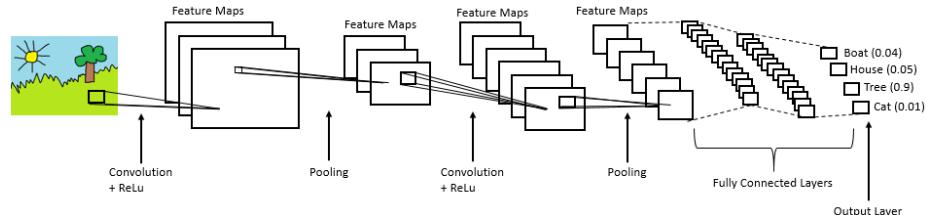
características obtenidas anteriormente a un vector y se alinea en una capa completamente conectada a una red neuronal (**Figura 36**).



**Figura 36.** Aplanado de matrices luego de agrupar la capa.

**Fuente:** (Prabhu, 2018).

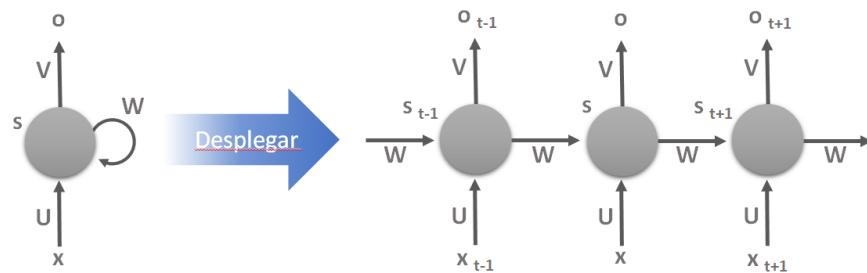
Para concluir, se muestra a continuación (**Figura 37**) la representación de la arquitectura completa de una Red Neuronal Convolucional resumiendo los conceptos anteriores.



**Figura 37.** Arquitectura completa de una CNN.

**Fuente:** (Prabhu, 2018).

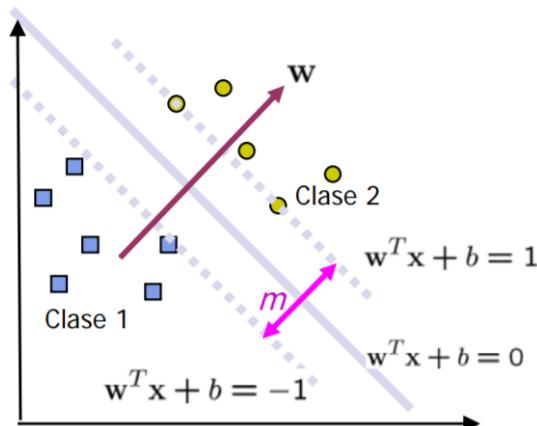
- ❖ **Redes Neuronales Recurrentes (RNN):** También conocidas por su nombre en inglés *Recurrent Neural Networks*, se caracterizan por no tener una estructura de capas como se aprecia en la **Figura 38**, sino más bien por permitir conexiones entre sus neuronas de manera arbitraria para crear temporalidad y que toda la red obtenga memoria. Todo esto permite generar una red muy potente para el análisis de secuencias, entre algunos ejemplos se mencionan el análisis de textos, sonidos o video (Calvo, Definición de Red Neuronal Recurrente, 2018).



**Figura 38.** Ejemplo de red neuronal recurrente.

**Fuente:** (Calvo, Definición de Red Neuronal Recurrente, 2018).

- ✓ **Máquina de Vectores de Soporte (SVM):** Es un algoritmo usado para tareas de regresión y clasificación, buscando un hiperplano en un espacio N-dimensional que clasifique claramente los puntos de datos a partir de la distancia máxima entre los puntos de datos de ambas clases. Para ello, maximiza la distancia del margen proporcionando cierto refuerzo para que los puntos de datos futuros puedan clasificarse con más confianza, es decir, que permita distinguir claramente dos clases, como se muestra en la **Figura 39** (Gandhi, 2018).



**Figura 39.** Hiperplano con dos clases separadas por una distancia  $m$ .

**Fuente:** (Betancourt, 2005).

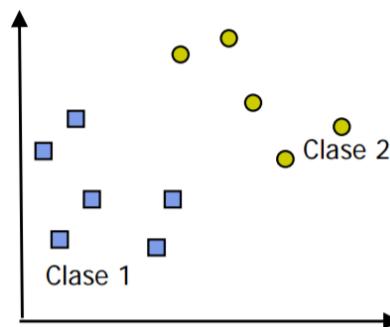
Este algoritmo tiene sus orígenes en la década de los años 60 en Rusia, desarrollados por Vapnik y Chervonenkis. Inicialmente se enfocó en el reconocimiento óptico de caracteres (OCR). Más tarde, los clasificadores de Vectores de Soporte se volvieron competitivos con los mejores sistemas disponibles en ese momento para resolver no solamente el anterior tipo de problema, sino también abarcar tareas de reconocimiento de objetos. En 1998,

se publicó el primer manual de estos algoritmos por Burges. Y debido a sus grandes resultados obtenidos en la industria, actualmente se usa con frecuencia en el campo del aprendizaje automático (Smola & Schölkopf, 2004).

Los vectores de soporte hacen referencia a un pequeño subconjunto de las observaciones de entrenamiento que se utilizan como soporte para la ubicación óptima de la superficie de decisión (MathWorks).

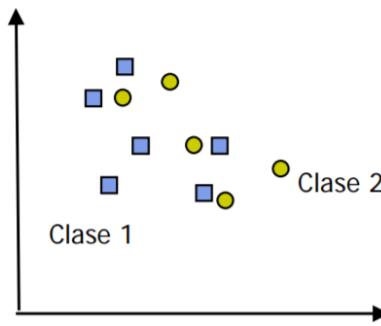
Una Máquina de Vectores de Soporte aprende la superficie decisión de dos clases distintas de los puntos de entrada. Como un clasificador de una sola clase, la descripción dada por los datos de los vectores de soporte es capaz de formar una frontera de decisión alrededor del dominio de los datos de aprendizaje con muy poco o ningún conocimiento de los datos fuera de esta frontera. Los datos son mapeados por medio de un *kernel* Gaussiano u otro tipo de *kernel* a un espacio de características en un espacio dimensional más alto, donde se busca la separación máxima entre clases. Cuando es traída de regreso al espacio de entrada, la función de frontera puede separar los datos en todas las clases distintas, cada una formando un agrupamiento. Esta teoría se basa en la idea de minimización de riesgo estructural (SRM), demostrando en muchas aplicaciones tener mejor desempeño que otros algoritmos de aprendizaje tradicional como las redes neuronales para resolver problemas de clasificación (Betancourt, 2005).

Cabe mencionar que hay casos en que el conjunto de datos de dos clases puede ser separables no necesariamente de forma lineal. En las **Figura 40** y **Figura 41** se observan casos linealmente y no linealmente separables, respectivamente.



**Figura 40.** Ejemplo de caso linealmente separable.

**Fuente:** (Betancourt, 2005).



**Figura 41.** Ejemplo de caso no linealmente separable.

**Fuente:** (Betancourt, 2005).

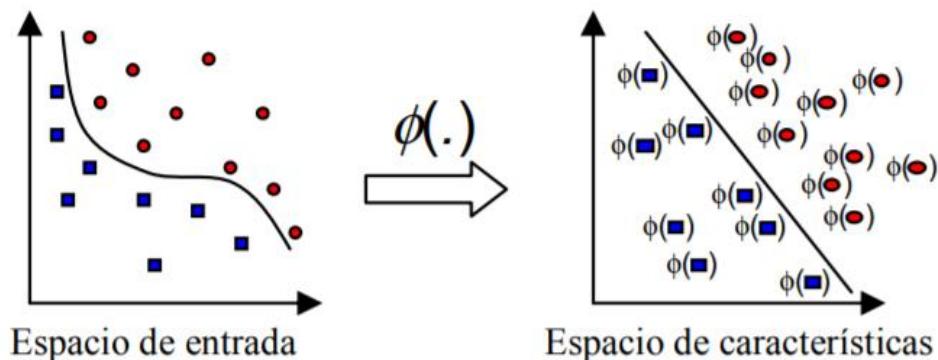
Lo que se debe hacer para el primer caso es crear el hiperplano a través de una función lineal  $w * z + b = 0$  y, definido el par  $(w, b)$ , separar el punto  $x_i$  según la función:

$$f(x_i) = \text{sign}(w * z + b) = \begin{cases} 1 & y_i = 1 \\ -1 & y_i = -1 \end{cases}$$

**Ecuación 20.** Ecuación del hiperplano para clasificar dos clases.

**Fuente:** (Betancourt, 2005).

Para el segundo caso, debido a su mayor complejidad, se puede introducir algunas variables no-negativas a la función del hiperplano para hallar su valor óptimo; o también es viable utilizar una función *kernel* que calcule el producto punto de los puntos de entrada en el espacio de características Z, como se aprecia en la **Figura 42**.



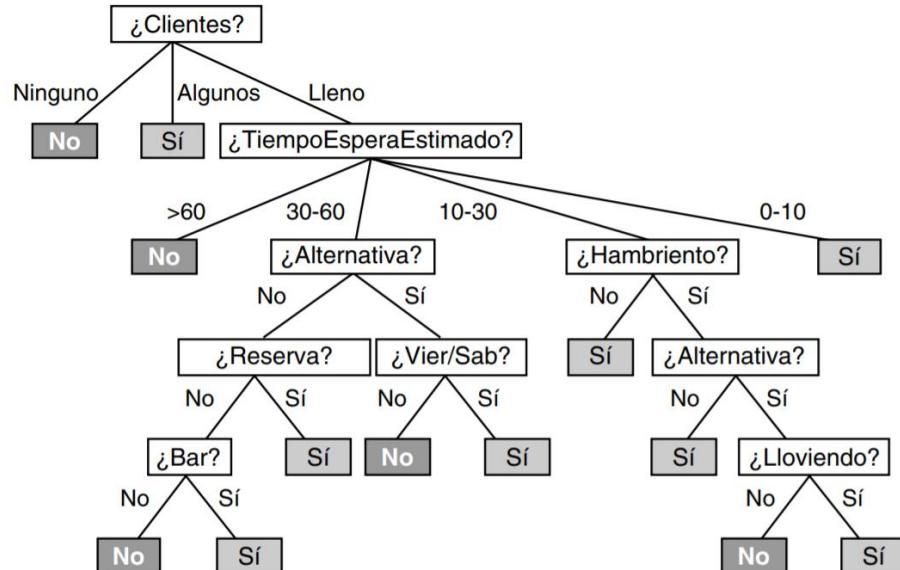
**Figura 42.** Aplicación de un kernel para transformar el espacio de los datos.

**Fuente:** (Betancourt, 2005).

- ✓ **Árboles de Decisión:** Representación visual de decisiones y toma de decisiones utilizada en la minería de datos para derivar una estrategia y alcanzar un objetivo particular. Se dibuja boca abajo con su raíz en la parte superior. Consta de nodos internos, los cuales se subdividen en ramas o bordes y su contenido, las hojas o decisiones (Gupta, 2017).

Un árbol de decisión toma como entrada un objeto descrito a través de un conjunto de atributos y devuelve una “decisión”. Estos pueden ser discretos o continuos. La salida puede tomar cualquiera de estos dos tipos de valores; en el caso que aprenda una función tomando valores discretos se le denominará clasificación, y en el caso que la función sea continua será llamada regresión. En las clasificaciones booleanas, es decir de dos valores o binaria, clasificará como verdadero (positivo) o falso (negativo). Para alcanzar una decisión, el árbol desarrolla una serie de pruebas a través de sus nodos y las ramas que salen del nodo son etiquetadas con los valores posibles de dicha propiedad. Además, cada nodo hoja del árbol representa el valor que ha de ser devuelto si es alcanzado (Russell & Norvig, Inteligencia Artificial: Un Enfoque Moderno, 2004).

Por ejemplo, representando un ejemplo de este algoritmo, se ilustra en la **Figura 43** para decidir si se debe esperar por una mesa en un restaurante.



**Figura 43.** Ejemplo del algoritmo de árbol de decisión.

**Fuente:** (Russell & Norvig, Inteligencia Artificial: Un Enfoque Moderno, 2004).

## 2.3. Marco Conceptual

### 2.3.1. Crowdfunding

El financiamiento colectivo o crowdfunding es un sistema que, utilizando internet como base de operaciones, busca generar una respuesta económica activa en el usuario (López-Golán, Vaca Tapia, Benavides García, & Coronado Otavallo, 2017).

Combinando ideas de microfinanzas y crowdsourcing, el crowdfunding es la práctica de financiar una empresa o un proyecto al recaudar muchas pequeñas cantidades de dinero de un gran número de personas. Este mecanismo de financiamiento ha sido recibido por los recaudadores de fondos, el público en general y los formuladores de políticas. La industria de crowdfunding ha crecido rápidamente en los últimos años, así como el crecimiento exponencial del número de plataformas de crowdfunding, el número de proyectos expuestos allí y el número de capital total recaudado en esos sitios web (Xuefeng & Zhao, 2018).

Existen 4 diferentes modelos de crowdfunding: basado en donaciones, basado en recompensas, basado en capital social y basado en deuda. El crowdfunding basado en capital social se encuentra actualmente limitado en los Estados Unidos debido a que la Regulación D de la Ley de Valores de 1933 prohíbe la participación de muchos potenciales inversionistas y los obliga a tener un ingreso anual mayor a \$ 200,000 o más de \$1 millón en patrimonio neto (Lichtig, 2015).

### 2.3.2. Kickstarter

Kickstarter, desde su inicio en 2009, es una plataforma de financiamiento de proyectos creativos de todo tipo, los cuales incluyen películas, juegos, música, arte, diseño y tecnología. Actualmente, se han registrado más de 162 mil proyectos realizados, 16 millones de contribuyentes y 4,3 miles de millones de dólares fondeados (Kickstarter). La plataforma utiliza un modelo de financiamiento llamado “todo o nada”, el cual consiste en que, si un proyecto no alcanza su meta de financiamiento en un determinado plazo de tiempo, no se realiza ninguna transacción de fondos (Kickstarter). Si bien los patrocinadores apoyan estos proyectos por motivos personales y distintos para hacerlos realidad, ellos no obtienen la propiedad o los ingresos de los proyectos que financian, sino que los creadores conservan la totalidad de su trabajo (Kickstarter).

### **2.3.3. Proyecto**

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. La naturaleza temporal de los proyectos indica un principio y un final definidos, y que el propósito se alcanza cuando se logran los objetivos del proyecto o cuando este es terminado por no cumplir sus objetivos, o cuando ya no existe la necesidad inicial que dio origen al mismo (Project Management Institute, 2017).

A partir de este concepto enunciado por el PMI, se entiende que un proyecto parte de una idea que un individuo o conjunto de individuos tienen en mente para convertirla en realidad con el fin de responder a una necesidad.

En Kickstarter, los proyectos que pueden ser financiados deben pertenecer a las siguientes categorías: Arte, Cómics, Artesanías, Baile, Diseño, Moda, Cine y vídeo, Comida, Juegos, Periodismo, Música, Fotografía, Publicaciones, Tecnología, y Teatro. Cualquier tipo de persona puede contribuir al mismo desde ser patrocinadores hasta formar parte de las principales referencias (Kickstarter).

Dentro de las normas internas de la plataforma para la creación de proyectos se especifica que cada proyecto debe resultar ser totalmente nuevo, original e innovador que pueda ser compartido con el público, así como haber sido presentados de forma honesta y clara. En adición a esto, también se menciona que las recaudaciones que se darán no podrán ser otorgadas a obras benéficas sino cumplir con el objetivo de llevar a cabo el proyecto planteado, al igual que está prohibido asimismo del ofrecimiento de incentivos financieros como resultado (Kickstarter).

### **2.3.4. Campaña**

Una campaña puede abarcar diversos términos si es que se busca su concepto en fuentes confiables como la Real Academia Española debido al alcance y empleabilidad del mismo. Para el actual contexto, el término se refiere específicamente a la campaña publicitaria, la cual se define como una estrategia diseñada y ejecutada en diferentes medios para obtener objetivos de notoriedad, ventas y comunicación de una determinada marca o producto a través del uso de la publicidad (Cyberclick).

En la plataforma de Kickstarter, existen personas que tienen ideas y proyectos en mente, buscando financiarlos en el sitio web. Para lograr su objetivo y siguiendo la regla del “todo o nada”, estos individuos realizan eventos de promoción para atraer entre personas conocidas suyas y cibernautas en general. A esta serie de eventos de promoción se le conoce como campañas. Las campañas exitosas se basan en que un determinado proyecto alcanza a ser financiado en el tiempo estimado gracias a algunas de las siguientes características (Kickstarter):

- ✓ Logran ser claros y concisos sobre lo que su proyecto busca alcanzar al ser financiado. Se definen bien las características del proyecto mediante detalles en la descripción, videos e imágenes precisas.
- ✓ Indican los beneficios y recompensas que los patrocinadores obtendrán si es que la campaña es exitosa y el proyecto se financia.
- ✓ Atrae e interactúa con una gran cantidad de público, manteniendo constante comunicación acerca de actualizaciones y novedades de la campaña y resolviendo dudas que puedan darse.
- ✓ Se estiman costos de manera eficiente que pueden ser solventados para cubrir más allá del proyecto una vez financiado, incluido los que se originan para la entrega del producto a los patrocinadores. Se logra convencerlos.

### 3. CAPÍTULO III: METODOLOGÍA DE LA INVESTIGACIÓN

#### 3.1. Diseño de la Investigación

##### Enfoque de la Investigación

El presente trabajo tendrá un enfoque cuantitativo ya que se busca diseñar y desarrollar instrumentos, en este caso modelos predictivos, para responder al problema estudiado a partir de medición de datos históricos en la plataforma Kickstarter con herramientas basadas en la estadística y matemáticas que puedan ser interpretadas por cualquier investigador (Hernández Sampieri, 2010).

##### Alcance de la Investigación

El alcance del presente trabajo será descriptivo ya que se recolectarán datos en un determinado rango de tiempo (desde 2009 hasta el presente año 2019) para describir el comportamiento de las campañas de proyectos tecnológicos en Kickstarter a partir de las características de sus variables y con ello, pronosticar su posible éxito o fracaso antes de finalizar la campaña con un nivel óptimo de precisión (Hernández Sampieri, 2010).

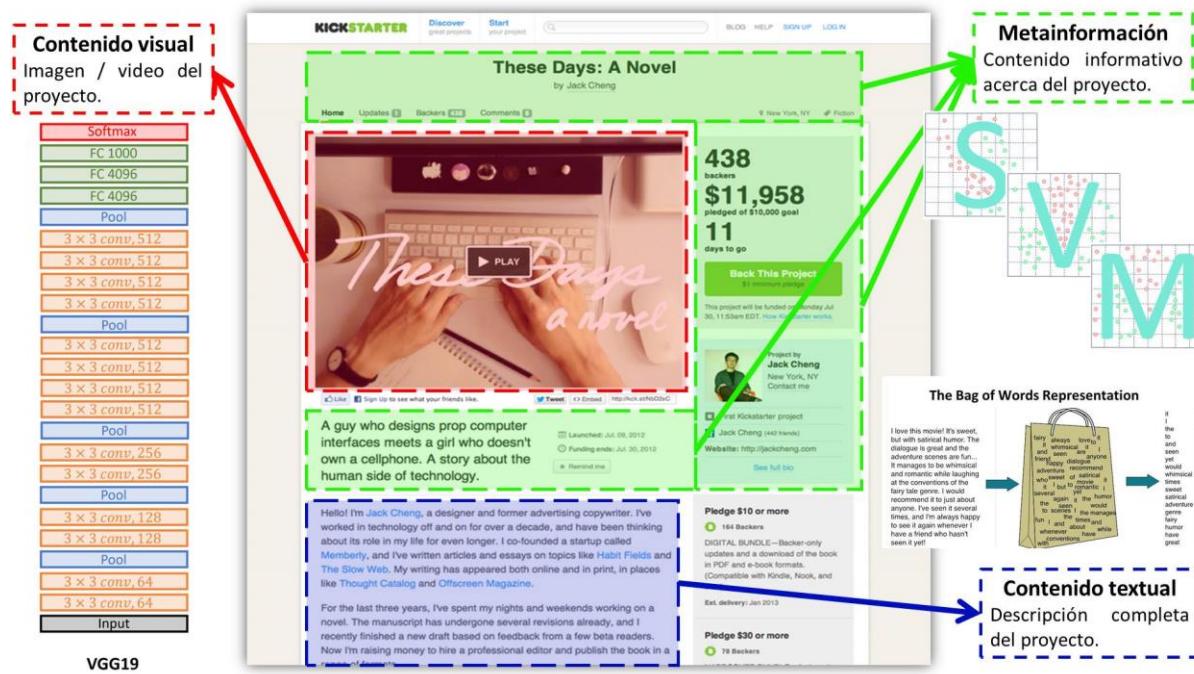
##### Tipo de la Investigación

Para determinar el tipo de la investigación, primero es necesario definir el actual trabajo como **Diseño Experimental** ya que las variables que se tienen serán controladas, es decir, serán agregadas o quitadas en el o los modelos construidos en el experimento para analizar el impacto que este o estos tendrán en los resultados obtenidos. Dentro de esta categoría se clasifica como **Diseño Experimental Puro** ya que se busca medir la variable dependiente, en este caso Status (el estado actual del proyecto en Kickstarter) a partir de la manipulación de las demás variables independientes agregando o desagregándolas para comparar los rendimientos obtenidos de los instrumentos de medición y determinar cuáles de ellas finalmente serán tomadas en cuenta (Hernández Sampieri, 2010).

##### Descripción del prototipo de Investigación

Teniendo como referencia y base principal el décimo antecedente explicado en el Capítulo II, la idea del prototipo final consistió en ensamblar las tres partes básicas de un proyecto: la primera consiste en el tratamiento de la metainformación (en la cual se realizarán, asimismo, tres experimentos independientes), el segundo, en

el contenido visual y el último, el contenido textual respectivamente pero con el valor diferenciado de adaptar el modelo general de acuerdo a las variables y conjuntos de datos disponibles para el presente trabajo. Para ello, se representa cada una de las tres partes agrupadas en el marco de trabajo de la **Figura 44**.



**Figura 44.** Marco de trabajo del prototipo final.

**Fuente:** Elaboración propia.

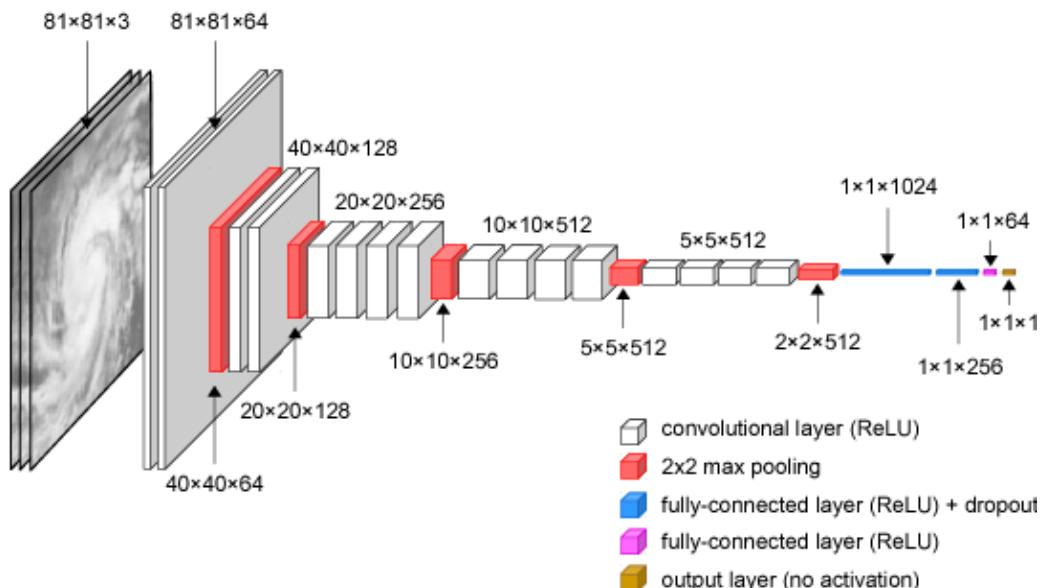
La metainformación es el contenido más completo acerca del proyecto. Contiene muchas variables desde la información demográfica y datos del o los autores y su proyecto, hasta los objetivos monetarios que se busca obtener. En todos los proyectos de Kickstarter, esta información contiene el título del proyecto, un slogan o “propaganda” que sea fácil de recordar para el público en general, su localización, datos de su(s) creador(es) y otras campañas lanzadas previamente, el número de patrocinadores que decidieron invertir hasta el deadline del proyecto, el monto meta que se espera alcanzar y su tipo de moneda (por lo general, USD), el monto total recaudado, días para que finalice la campaña (si es que sigue vigente), fechas de creación, lanzamiento y finalización, la duración en días (si es que ya finalizó), actualizaciones y comentarios que se realizaron, y el estado final que alcanzó la campaña (si es que no finaliza aún, no se muestra).

Para la metainformación, se seleccionaron las variables más relevantes siguiendo como base los antecedentes del **Capítulo II**, se normalizaron y

estandarizaron sus variables independientes y, luego de fraccionar el conjunto total de datos en entrenamiento y prueba, se realizó dos experimentos: el primero entrenando un modelo de Máquina de Vectores de Soporte (SVM) fraccionados en 0.80 para el conjunto de entrenamiento y 0.20 para el de prueba; mientras que en el segundo experimento se usó un modelo de Red Neuronal con las mismas proporciones.

El contenido visual contiene la imagen o video principal del proyecto que se muestra por debajo del título y la propaganda del mismo. El fin de estos es mostrar al público el prototipo final del producto o servicio que se ofrecerá a los patrocinadores una vez alcanzado el financiamiento meta. Por ello, es muy importante que este sea atractivo, resuma todas sus características y explique qué beneficios tendrán aquellos que opten por invertir en la campaña. La mayoría de proyectos en Kickstarter tienen imágenes rectangulares de 315 píxeles de alto y 560 de ancho.

Para el contenido visual, se usó una arquitectura de red neuronal convolucional llamada VGG-19, un derivado del Visual Geometry Group (VGG) con 19 capas. Su arquitectura es similar al VGG-16 creado por K. Simonyan y A. Zisserman de la Universidad de Oxford, especializado en el reconocimiento de imágenes a gran escala, incluso alcanzando el 92.7% de precisión de prueba top 5 en ImageNet (ul Hassan, 2018). Sin embargo, y a diferencia del modelo propuesto en el antecedente, se opta por esta red por su mejor performance gracias a sus 3 capas adicionales intermedias.



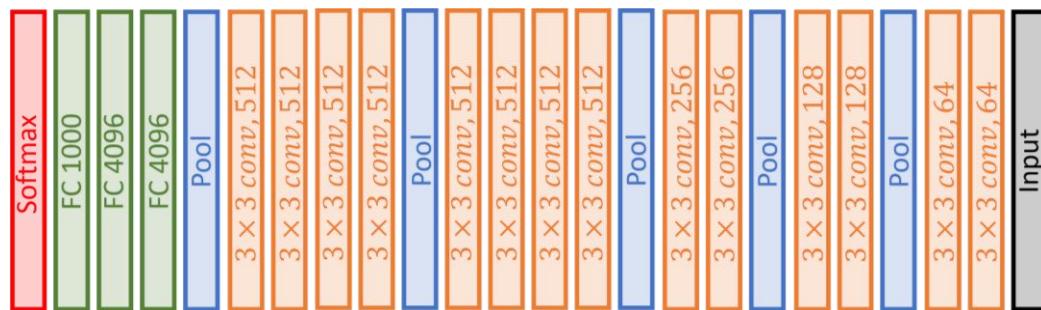
**Figura 45.** Arquitectura de un modelo de red VGG-19.

**Fuente:** (Combinido, Mendoza, & Aborot, 2018).

Finalmente, el contenido textual contiene la descripción completa del proyecto, desde el origen de la idea hasta las características y beneficios que el producto o servicio ofrecerá para todos aquellos que decidan invertir en el mismo. La descripción busca dar soporte al contenido visual (imagen o vídeo) ubicado encima. Asimismo, detalla el proceso que se siguió en la campaña antes de que sea creada. Por ello, muchas personas buscan obtener respuestas que tal vez no fueron abordados en la imagen o video resumen del proyecto.

Para el contenido textual, se usaron los modelos de **Bolsa de Palabras** (*Bag of Words*, BoW) y **Frecuencia de Término-Frecuencia Inversa de Documento** (*Term Frequency–Inverse Document Frequency*, TF-IDF) para contabilizar todas las palabras únicas (luego de eliminar las que no presentan sintagma o significado conocidas como “*stop words*”) a partir de un vocabulario de palabras etiquetadas o *tokenizadas*. Estos son muy útiles para conocer las palabras más frecuentes en un documento que puedan y que influyan en el estado final. Posteriormente, se creó un modelo de Máquina de Vectores de Soporte para evaluar su precisión de ambos.

Una vez obtenido los mejores resultados de las tres partes del marco de trabajo, se pretendió ensamblarlos, mediante Transfer Learning, de la siguiente forma: las características obtenidas del contenido textual y la metainformación son añadidas en las últimas 3 capas del modelo VGG-19 del contenido visual, que representan a su vez las capas totalmente conectadas, como se presenta en la **Figura 46**.



**Figura 46.** Arquitectura detallada de un modelo de red VGG-19.

**Fuente:** (DataHacker, 2018).

### 3.2. Población y muestra

#### Población

La población que será considerada para el presente trabajo será de 27,251 proyectos en Kickstarter de la categoría tecnología de todas las subcategorías entre los períodos 2009-2019, en su mayoría del territorio de los Estados Unidos de América.

## **Muestra**

Debido a que se 214 imágenes del contenido visual no pudieron re-dimensionarse, así como 2 proyectos no contaban con descripciones en el contenido textual, se procedió a remover los 216 proyectos incompletos tanto en la metainformación como en las otras bases de datos, resultando finalmente en 27,035 registros en cada una de los tres conjuntos de datos. Sin embargo, la división en subconjuntos fue distinta en los tres casos y se dio de la siguiente manera:

- Para la metainformación y el contenido textual, el conjunto de datos total de cada uno fue dividido en un subconjunto de entrenamiento (80%) y uno de prueba (20%) siguiendo las proporciones dadas en el octavo antecedente.

Conjunto de datos	Total	%
<b>Entrenamiento</b>	21,628	80
<b>Prueba</b>	5,407	20

**Tabla 3.** Distribución de los registros para el segundo experimento.

**Fuente:** Elaboración propia.

- Para el contenido visual, el conjunto de datos total fue dividido en tres subconjuntos: entrenamiento (80%), validación (10%) y prueba (10%) siguiendo las proporciones dadas en el décimo antecedente.

Conjunto de datos	Total	%
<b>Entrenamiento</b>	21,628	80
<b>Validación</b>	2,703	10
<b>Prueba</b>	2,704	10

**Tabla 4.** Distribución de los registros para el tercer experimento.

**Fuente:** Elaboración propia.

## **Unidad de Análisis**

La unidad de análisis para el presente trabajo será un proyecto en Kickstarter de la categoría tecnología de cualquier subcategoría entre los períodos 2009-2019 dentro del territorio de los Estados Unidos de América.

### 3.3. Operacionalización de variables

En la **Tabla 5** se presentan las variables a usar para el conjunto de datos final basado en contenido textual y metainformación. Estas fueron seleccionadas de acuerdo al Benchmarking aplicado a los 10 antecedentes en el Capítulo II.

Los autores citados son los siguientes:

- ✓ (Cheng, Tan, Hou, & Wei, 2019)
- ✓ (Jin, Zhao, Chen, Liu, & Ge, 2019)
- ✓ (Kamath & Kamat, 2018).
- ✓ (Kaur & Gera, 2017).
- ✓ (Li, Rakesh, & Reddy, 2016).
- ✓ (Yu, y otros, 2018).
- ✓ (Yuan, Lau, & Xu, 2016).
- ✓ (Zhou, y otros, 2015).

Asimismo, para el contenido visual se descargaron las fotos de los más de 27,000 proyectos en formato .png. El objetivo es seguir la aplicación del décimo antecedente (Cheng, Tan, Hou, & Wei, 2019) a partir de una Red Neuronal Multicapa que alimente el modelo general y mejore su poder predictivo.

Columna	Descripción	Tipo de Dato
<b>Id</b>	Identificador interno en Kickstarter.	int64
<b>backers_count</b>	Número de patrocinadores.	int64
<b>Name</b>	Nombre del proyecto	object
<b>Blurb</b>	Propaganda del proyecto.	object
<b>Category</b>	Subcategoría del proyecto.	object
<b>Description</b>	Descripción del proyecto.	object
<b>Photo</b>	URL de la foto del proyecto.	object
<b>Urls</b>	Enlace URL del proyecto.	object
<b>City</b>	Ciudad de origen del proyecto.	object
<b>Country</b>	País de origen del monto prometido.	object
<b>Goal</b>	Meta de financiamiento.	float64
<b>Pledged</b>	Monto prometido (en moneda original).	int64
<b>Currency</b>	Divisa o moneda del monto prometido.	object
<b>usd_pledged</b>	Monto prometido (equivalente en dólares americanos).	int64
<b>created_at</b>	Fecha de creación del proyecto.	int64
<b>launched_at</b>	Fecha de lanzamiento del proyecto.	int64
<b>Deadline</b>	Fecha límite del proyecto.	int64
<b>Duration</b>	Duración del proyecto en días (deadline-launched_at).	int64
<b>State</b>	Condición actual del proyecto.	object

**Tabla 5.** Descripción de las variables a usar para el conjunto de datos final.

**Fuente:** Elaboración propia.

Donde el tipo de datos abarca registros numéricos (int64 y float64), categóricas (object) y binarias (bool).

Con el nuevo conjunto de datos se puede visualizar la distribución que cada variable sigue, así como algunas medidas estadísticas mediante las librerías que Python ofrece como matplotlib, numpy, entre otras.

La variable independiente target será **status**, ya que representa la condición actual o estado en el que el proyecto se encuentra (cancelado, fallido, vivo, exitoso, suspendido); mientras que el resto de columnas del conjunto de datos representan las variables dependientes que determinarán su performance en el modelo. Si se detectase mediante gráficos o plot que la proporción de valores de la variable target, es decir si

tuvo éxito (1) o fracaso (0), están demasiado desbalanceadas o presenta una proporción mayor a 70%-30%, se procederá a realizar el balanceo mediante la técnica de Over-Sampling con la función SMOTE, la cual incrementará el número de registros a partir de datos sintéticos de aquel valor, sea éxito o fracaso, que presente el menor número de registros en la proporción hasta llegar a un balance equitativo, es decir, 50%-50%.

### 3.4. Instrumentos de medida

Los instrumentos de medida que servirán para determinar la performance del o los modelos construidos en el experimento serán algunas de las métricas de clasificación de Machine Learning descritas en los papers tomados como antecedentes en el Capítulo II del presente trabajo y seleccionadas mediante Benchmarking considerando como criterio la similitud del problema y propuestas de solución más aproximadas. Antes de proceder a explicar detalladamente cada una de ellas, es necesario conocer los conceptos de la Matriz de confusión, así como sus elementos que la componen.

- ✓ **Matriz de confusión:** Es una tabla de NxN que resume el nivel de éxito de las predicciones de un modelo de clasificación; es decir, la correlación que existe entre la etiqueta y la clasificación del modelo. Un eje de una matriz de confusión es la etiqueta que el modelo predijo; el otro es la etiqueta real. N representa el número de clases. Es un problema de clasificación binaria, N=2 (Kohavi & Provost, 1998). Su principal objetivo es describir el rendimiento de un modelo supervisado de Machine Learning en los datos de prueba, donde se desconocen los verdaderos valores. Se le llama “matriz de confusión” porque hace que sea fácil detectar dónde el sistema está confundiendo dos clases (SitioBigData.com, 2019). Se representa de la siguiente manera:

		Valores Actuales	
		Positivos (1)	Negativos (0)
Valores Predichos	Positivos (1)	Verdaderos Positivos (VP)	Falsos Positivos (FP)
	Negativos (0)	Falsos Negativos (FN)	Verdaderos Negativos (VN)

**Tabla 6.** Matriz de Confusión.

**Fuente:** (Izco, 2018).

De la anterior tabla, existen 4 elementos clave:

- ✓ **Verdadero positivo** (TP o *True positive*): Es el ejemplo en el que el modelo predijo de manera correcta la clase positiva. Por ejemplo, el modelo infirió correctamente que un paciente con determinadas características descritas en las variables sufre de cáncer (Google Developers, 2018).
- ✓ **Verdadero negativo** (TN o *True negative*): Es el ejemplo en el que el modelo predijo de manera correcta la clase negativa. Por ejemplo, el modelo infirió correctamente que una determinada especie animal de acuerdo a sus características no era un mamífero (Google Developers, 2018).
- ✓ **Falso positivo** (FP, *false positive* o Error del Tipo I): Es el ejemplo en el que el modelo predijo de manera incorrecta la clase positiva. Por ejemplo, el modelo infirió que un paciente varón presentaba embarazo (clase positiva) cuando en realidad no era así (Google Developers, 2018).
- ✓ **Falso negativo** (FN, *false negative* o Error del Tipo II): Es el ejemplo en el que el modelo predijo de manera incorrecta la clase negativa. Por ejemplo, el modelo infirió que un mensaje de correo electrónico en particular no era spam (clase negativa), pero ese mensaje en realidad sí era spam (Google Developers, 2018).

Explicado los conceptos anteriores, se derivan las siguientes métricas de clasificación usadas comúnmente, de las cuales serán usadas solo las 3 primeras tomando como referencia los papers de los antecedentes:

- ✓ **Exactitud** (*accuracy*): Representa la fracción de predicciones que se realizaron correctamente sobre el total de ejemplos en un modelo de clasificación. Se determina mediante la siguiente fórmula (Kohavi & Provost, 1998):

$$\text{Exactitud} = \frac{\text{Verdaderos positivos} + \text{Verdaderos negativos}}{\text{Verdaderos positivos} + \text{Verdaderos negativos} + \text{Falsos positivos} + \text{Falsos negativos}}$$

**Ecuación 21.** Fórmula para calcular la exactitud.

**Fuente:** (Kohavi & Provost, 1998).

Esta métrica responde a la pregunta ¿Cuál es la proporción de predicciones que se realizaron correctamente? (Izco, 2018).

- ✓ **Precisión (precision):** Representa el número de elementos identificados correctamente como positivo de un total de elementos identificados como positivos (SitioBigData.com, 2019). Se calcula mediante la siguiente fórmula:

$$\text{Precisión} = \frac{\text{Verdaderos positivos}}{\text{Verdaderos positivos} + \text{Falsos positivos}}$$

**Ecuación 22.** Fórmula para calcular la precisión.

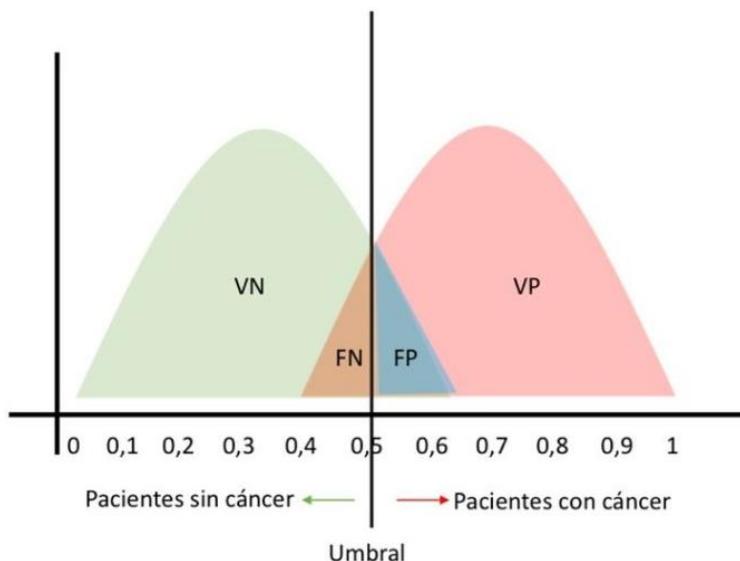
**Fuente:** (Kohavi & Provost, 1998).

Esta métrica responde a la pregunta ¿Qué proporción de predicciones positivas es correcta? (Izco, 2018).

- ✓ **Área bajo la curva ROC (AUC):** Considera todos los umbrales de clasificación posibles. Representa la probabilidad de que un clasificador tenga más seguridad de que un ejemplo resulte ser un verdadero positivo con respecto a que sea un falso positivo (Google Developers, 2018). Para entender el concepto del área, se necesita entender qué es la curva ROC y para qué sirve en primer lugar.

La curva ROC permite cuantificar la performance de distinción entre dos cosas del modelo como, por ejemplo, si un paciente tiene cáncer o no.

Siguiendo el anterior ejemplo, se tiene un modelo que predice si un paciente sufre de cáncer o no, cuyo resultado es el siguiente (**Figura 47**):



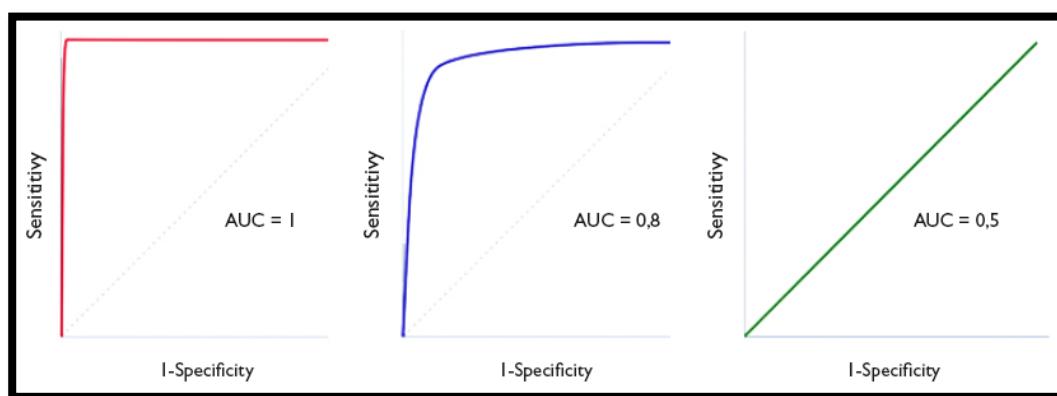
**Figura 47.** Descripción de resultados de modelo descriptivo de ejemplo.

**Fuente:** (González, 2019)

En la Figura N° 46 se puede observar que el área de borde verde (que contiene a los Falsos Positivos y el total de Negativos) representa a todos los pacientes que no tienen cáncer, mientras que el área de borde rojo (que contiene a los Falsos Negativos y el total de Positivos) representa a todos los pacientes que sí tienen cáncer. El umbral, que está establecido con valor 0.5, representa el punto de corte en el que el modelo clasificará a todos los pacientes por encima de ese valor como positivos, es decir, que sí tienen cáncer; mientras que aquellos por debajo del valor del umbral serán clasificados como negativos, es decir, que no tienen cáncer.

Cuando el umbral se desplaza hacia la izquierda, es decir, cuando la sensibilidad aumenta, la especificidad disminuirá. Por el contrario, cuando el umbral se desplaza hacia la derecha, la sensibilidad disminuirá y la especificidad aumentará. Se concluye entonces que existe una relación inversa entre la sensibilidad y la especificidad. En la curva ROC se representa la sensibilidad (1-especificidad) (González, 2019).

Ahora bien, el área que se grafica bajo esta curva explicará qué tan bien funciona el modelo. Este tendrá un mejor desempeño si la curva se aleja de la diagonal principal como se observa en la **Figura 48**.



**Figura 48.** Comparación de tres resultados de la curva AUC en el modelo.

**Fuente:** (Molina Arias & C, 2017)

Una interpretación básica del área bajo la curva ROC respecto del poder discriminante del modelo se muestra a continuación (Britos, García Martínez, Hossian, & Sierra, 2006):

- Si el área bajo la curva ROC = 0.5, entonces el poder discriminante del modelo es nulo.
- Si el área bajo la curva  $0.5 < \text{ROC} < 0.7$ , entonces el poder discriminante del modelo no es aceptable.

- Si el área bajo la curva  $0.7 \leq \text{ROC} < 0.8$ , entonces el poder discriminante del modelo es aceptable.
  - Si el área bajo la curva  $0.8 \leq \text{ROC} < 0.9$ , entonces el poder discriminante del modelo es excelente.
  - Si el área bajo la curva  $\text{ROC} \geq 0.9$ , entonces el poder discriminante del modelo es excepcionalmente bueno.
- ✓ **Sensibilidad** (*recall, sensitivity o True Positive Rate*): Representa el número de elementos correctamente identificados como positivos del total de positivos verdaderos (SitioBigData.com, 2019). Se calcula mediante la siguiente fórmula:

$$\text{Sensibilidad} = \frac{\text{Verdaderos positivos}}{\text{Verdaderos positivos} + \text{Falsos negativos}}$$

**Ecuación 23.** Fórmula para calcular la sensibilidad.

**Fuente:** (Kohavi & Provost, 1998).

- ✓ **Puntaje F1** (*F1 score*): Representa la media armónica de la precisión y la sensibilidad. Normalmente, se usa cuando uno difiere mucho del otro y no es posible realizar una conclusión determinante ya que solo es posible predecir bien una clase (SitioBigData.com, 2019). Se calcula mediante la siguiente fórmula:

$$\text{Puntaje F1} = \frac{2 * \text{Precisión} * \text{Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}}$$

**Ecuación 24.** Fórmula para calcular el puntaje F1.

**Fuente:** (Kohavi & Provost, 1998).

### 3.5. Técnicas de recolección de datos

Los datos recolectados para el trabajo son un mix de cuantitativos en su mayoría como por ejemplo el monto prometido, y cualitativos que incluye variables categóricas como la descripción del proyecto. Se recolectaron los datos a través de bases de datos públicas disponibles en Internet como el de la página Web Robots, fundada por los ex corporativos de TI Tomás Vitulskis y Paulius Jonaitis, la cual basada en dos plataformas en la nube ejecutadas una vez al mes, recolecta datos web mediante scraping para empresas (Web Robots). Para encontrar algunos de los papers con la información requerida más cercana, se utilizaron keywords o palabras clave como crowdfunding, Machine Learning, prediction, Kickstarter, accuracy y projects.

### **3.6. Técnicas para el procesamiento y análisis de la información**

Como se explicó en el punto 2.2.3 del Marco Teórico del presente trabajo, dentro de los sistemas de analítica de negocio, Big Data y Minería de Datos, tres de las metodologías más usadas son CRISP-DM, SEMMA y KDD (Braulio Gil & Curto Díaz, 2015).

Después de evaluar y comparar las tres mencionadas, se concluyó que la mejor de ellas dependerá de la intención o finalidad que busca lograr el o los involucrados en el negocio tanto en objetivos como en resultados. Para el presente trabajo de tesis, se seguirá la metodología de CRISP-DM debido a la coyuntura en la que se desarrollará. Las razones de la elección de la misma son las siguientes:

- ✓ La metodología CRISP-DM contempla entre sus fases la comprensión del negocio además de la parte técnica que incluye el modelado y análisis de resultados. La comprensión del negocio tiene un rol importante ya que se define al inicio de todo el proceso para dar el alcance del proyecto y definir objetivos que se buscan a partir de la Minería de Datos y Big Data.
- ✓ Ayuda a la gerencia del proyecto en la planeación y toma de decisiones (fase Despliegue) a partir de los resultados obtenidos, reportándolos y convirtiéndolos en oportunidades a considerar en los objetivos del negocio.
- ✓ Evalúa en todo el proceso los datos y variables usadas con el fin de crear el mejor modelo. Las variables seleccionadas serán importantes para interpretar los resultados y tomar decisiones.
- ✓ Si se habla de las otras dos metodologías, empezando por KDD, si bien esta contempla 9 pasos durante su proceso, el objetivo de KDD en cada uno de estos resulta ser más técnico, es decir, trabajar, seleccionar e interpretar métricas, variables, modelos, entre otros para obtener los mejores resultados más allá de considerar el contexto y comprensión del negocio. De hecho, no existe alguna fase dedicada al entendimiento del mismo.
- ✓ Por otra parte, la metodología SEMMA se basa, como su nombre lo indica, en la selección, exploración y modelado de grandes cantidades de datos para descubrir patrones de negocio desconocidos. Sin embargo, al limitarse a 5 fases comenzando con la fase de muestreo, no hace hincapié en la comprensión del negocio, sino más bien comienza con el procesamiento de datos para la construcción del modelo.

Luego de elegir la metodología a seguir en el presente proyecto, se procede a explicar en este punto 4 de las 6 fases de la metodología CRISP-DM. Las dos fases restantes serán explicadas en el punto 3.3.

### **Comprensión del negocio**

La primera fase de la metodología CRISP-DM consiste en la definición del problema de Minería de Datos que se tiene, así como el entendimiento de los objetivos y requerimientos que se espera lograr en el proyecto.

Para empezar, el crowdfunding se basa, como su nombre en inglés lo indica, en lograr que un proyecto emprendedor sea llevado a cabo gracias al financiamiento colectivo. Existen 4 diferentes modelos de crowdfunding: basado en donaciones, basado en recompensas, basado en capital social y basado en deuda. El crowdfunding basado en capital social se encuentra actualmente limitado en los Estados Unidos debido a que la Regulación D de la Ley de Valores de 1933 prohíbe la participación de muchos potenciales inversionistas y los obliga a tener un ingreso anual mayor a \$ 200,000 o más de \$1 millón en patrimonio neto (Lichtig, 2015).

El modelo basado en donaciones se refiere a la recaudación de fondos a través de la Web 2.0, en el cual los patrocinadores no esperan recompensas materiales a cambio sino más bien una recompensa social. Por el contrario, el modelo basado en recompensas ofrece compensación tanto material como inmaterial y representa hoy en día el modelo de crowdfunding más frecuente. Los financiadores, por un lado, pueden verse beneficiados de la venta anticipada, recibiendo el proyecto o producto financiado antes de su publicación o entrada al mercado al mejor precio. Los proyectos que pertenecen a esta categoría a menudo son organizaciones sin fines de lucro (Kraus, Richter, Brem, Cheng, & Chang, 2016).

Kickstarter, la plataforma de crowdfunding más citada, analizada y una de las más grandes, es una comunidad basada en el crowdfunding por recompensas. A la fecha, 128 mil proyectos han sido financiados, 3 billones de dólares fueron prometidos en proyectos y 13 millones de patrocinadores han participado, del cual el 31% ha colocado dinero en más de 1 proyecto.

La particularidad de los proyectos en esta plataforma es que, luego de cumplir un plazo de tiempo determinado, el monto prometido acumulado puede ser asignado a los proyectos que hayan alcanzado o superado la meta estipulada al inicio. Para ello, los dueños de los proyectos tienen que crear estrategias para lograr campañas exitosas y

hacer realidad su producto. Un 75% de los proyectos que fueron apoyados por al menos 25 patrocinadores han sido financiados exitosamente (Kickstarter).

Para lograr que un proyecto en la plataforma logre impactar en la comunidad, debe contar con conexiones duraderas, retroalimentación del producto y ser tangible. Asimismo, algunos enfoques importantes se basan en probar nuevas ideas, aumentar la comunidad y cubrir costos de manufacturación.

Para empezar a realizar la campaña del proyecto, es necesario seguir las siguientes sugerencias (Yu A. , 2017) y (Kickstarter):

- ✓ Identificar la meta óptima del proyecto. Se debe realizar una lista de todos los costos, así como el shipping que deben de cada país. También se debe contar con un buffer en caso ocurra un imprevisto. Se puede determinar la meta a través de la fórmula (Gastos + Buffer) x Alcance.
- ✓ Crear el video de la campaña en un presupuesto. El vídeo resulta ser un factor muy importante ya que aquellos que cuentan con un video de su proyecto tienen un 50% de chance de tener éxito. El promedio debería ser de 3.38 minutos, aunque el público presta más atención hacia aquellos que duran poco. El vídeo debe contener y responder las siguientes preguntas:
  - ✓ ¿Qué hace memorable al proyecto?
  - ✓ ¿Cómo se verá el proyecto?
  - ✓ ¿Por qué se está creando el producto?
  - ✓ ¿Por qué lo necesitas?
  - ✓ ¿Por qué es importante que tu proyecto exista?
  - ✓ ¿Por qué debes ser tú su creador?
  - ✓ ¿Por qué tu compañía?
  - ✓ ¿Cómo proveerás una solución?
  - ✓ ¿Cómo funciona la solución?

Es importante destacar las estadísticas del problema que se piensa resolver al inicio del video ya que el primer minuto resulta ser el lapso de tiempo que más capta la atención del público. Se recomienda también tener buen audio e iluminación.

- ✓ Escribir la descripción del proyecto. Otro factor clave en el desempeño de la campaña de un proyecto en Kickstarter resulta ser la descripción que representa un resumen de los puntos más relevantes del producto. Se debe mencionar una breve introducción del proyecto, cómo funciona, para quiénes son, el proceso que se siguió en el desarrollo, detalles y especificaciones, y un cronograma e hitos.

- ✓ Incluir imágenes sobre cómo lucirá el proyecto. Esta prueba convencerá a más de una persona en querer invertir en el proyecto. Las imágenes a veces atraen más a las personas que no gustan leer la descripción completa del proyecto.
- ✓ Determinar las recompensas del proyecto. Al basarse en un modelo de crowdfunding en recompensas, los patrocinadores esperan un beneficio por parte del proyecto una vez este logre alcanzar su financiamiento. El número ideal de recompensas puede ser entre 5 y 7. Una recompensa en promedio de \$100 es la que genera más dinero. Las recompensas pueden ser de cuatro tipos: el producto en sí mismo entregado hacia los patrocinadores antes que este salga al mercado, reconocimientos en la página web y acceso hacia actualizaciones, recuerdos y certificados de membresía, y nuevas experiencias como visita a los desarrolladores en sus estudios.

Todos los factores anteriores mencionados determinarán en gran medida el nivel de éxito que tendrá un proyecto una vez su campaña sea lanzada en la plataforma.

Líneas arriba en la Figura N° 2 se mencionó que los proyectos tecnológicos apenas alcanzan un aproximado de 25% de ser financiados. El objetivo de este trabajo además de crear un modelo predictivo con una mejor precisión que trabajos previos es determinar las causas por las cuales un bajo número de proyectos de esta categoría alcanzan el éxito.

### **Comprensión de los datos**

Una vez comprendido el funcionamiento del negocio y sus objetivos que busca lograr, la siguiente fase es entender el significado de los datos con los que se van a trabajar. Estos incluyen valores y variables que determinarán la performance del modelo que se elaborará en las siguientes etapas.

Se recolectó bases de datos de proyectos en Kickstarter de diferentes fuentes, entre ellas el repositorio web Kaggle. A continuación, se describirán las variables o columnas del conjunto inicial de datos recopilado en la **Tabla 7**:

Columna	Descripción	Tipo de Dato
<b>backers_count</b>	Número de patrocinadores.	int64
<b>Blurb</b>	Propaganda del proyecto.	object
<b>Category</b>	Subcategoría del proyecto.	object
<b>converted_pledged_amount</b>	Monto prometido convertido a moneda local.	int64
<b>Country</b>	País de origen del monto prometido.	object
<b>created_at</b>	Fecha de creación del proyecto.	int64
<b>Creator</b>	Creador del proyecto.	object
<b>Currency</b>	Código de moneda para apoyar proyecto.	object
<b>currency_symbol</b>	Símbolo de moneda para apoyar proyecto.	object
<b>currency_trailing_code</b>	Si la moneda tiene código final.	bool
<b>current_currency</b>	Actual moneda usada.	object
<b>Deadline</b>	Fecha límite del proyecto.	int64
<b>disable_communication</b>	Si el dueño del proyecto interactúa en la plataforma.	bool
<b>Friends</b>	Lista de amigos del dueño del proyecto.	object
<b>fx_rate</b>	Ratio desconocida.	float64
<b>Goal</b>	Meta de financiamiento.	float64
<b>Id</b>	Identificador interno en Kickstarter.	int64
<b>is_backing</b>	Si el proyecto está apoyado.	bool
<b>is_starrable</b>	Si el proyecto puede ser destacado.	bool
<b>is_starred</b>	Si el proyecto está destacado.	bool
<b>launched_at</b>	Fecha de lanzamiento del proyecto.	int64
<b>Location</b>	Localización del proyecto.	object
<b>main_category</b>	Categoría principal del proyecto.	object
<b>Name</b>	Nombre del proyecto.	object
<b>Permissions</b>	Permisos o licencias del proyecto.	object
<b>Photo</b>	Foto o imagen del proyecto.	object
<b>Pledged</b>	Monto prometido por patrocinadores.	float64
<b>Profile</b>	Perfil del creador del proyecto.	object
<b>Slug</b>	Título breve del proyecto.	object
<b>source_url</b>	Directorios web del proyecto.	object

<b>Spotlight</b>	Si el proyecto tiene Spotlight.	bool
<b>staff_pick</b>	Si el personal del proyecto fue seleccionado.	bool
<b>State</b>	Condición actual del proyecto.	object
<b>state_changed_at</b>	Fecha de actualización del estado del proyecto.	int64
<b>static_usd_rate</b>	Ratio estático de moneda (en dólares).	float64
<b>Urls</b>	Enlaces externos del proyecto.	object
<b>usd_pledged</b>	Monto prometido (en dólares).	float64
<b>usd_type</b>	Tipo de moneda usada.	object

**Tabla 7.** Descripción de las variables del conjunto de datos recolectado.

**Fuente:** (Zegarra García, 2018)

### **Preparación de los datos**

Una vez entendido el significado de cada variable, tipo de dato y determinación de valores faltantes o nulos, se procede a preparar un nuevo conjunto de datos que solo contenga aquellas variables representativas para la elaboración del modelo. De acuerdo a algunos de los papers y referencias mencionadas anteriormente en los antecedentes del Marco Teórico, de las 38 variables mostradas en la **Tabla 7** se seleccionarán 15 de ellas que representen las más significativas para el nuevo conjunto de datos, así como se excluirá aquellas que contengan bastantes valores faltantes o nulos. Asimismo, se añadirán variables que guarden relación con las conexiones de los dueños del proyecto en sus redes sociales.

En la Tabla N° 3 se presentaron las potenciales variables del conjunto de datos final.

### **Modelamiento**

Luego del pre-procesamiento de los datos, en esta fase se procederá a crear los modelos predictivos para resolver el problema. Tomando de referencia algunos de los papers en el Marco Teórico, se mencionan modelos de Aprendizaje Automático (Machine Learning) como Naive Bayes, Árboles de Decisión, Máquina de Vectores de Soporte, Random Forest, Regresión Logística, entre otras. Asimismo, también se implementará el modelo de Aprendizaje Profundo (Deep Learning) de Redes Neuronales Artificiales (RNA). Todos los modelos mencionados serán evaluados individual y colectivamente.

## **Evaluación**

En la quinta fase, cada modelo registrará distintos resultados tanto en métricas como velocidad y predicción de resultados. Se revisarán los procesos seguidos en cada uno y, de ser necesario, se establecerán pasos adicionales. Previamente al análisis de resultados, se deben haber evaluados la(s) mejor(es) métrica(s) luego de un proceso de selección. Algunas de las que se pueden mencionar son (SitioBigData.com, 2019):

- ✓ Exactitud.
- ✓ Precisión.
- ✓ Sensibilidad.
- ✓ Especificidad.
- ✓ Área bajo la curva de funcionamiento del receptor (ROC).
- ✓ F1-Score.
- ✓ Matriz de confusión.
- ✓ Pérdida logarítmica.
- ✓ Cohen's Kappa.

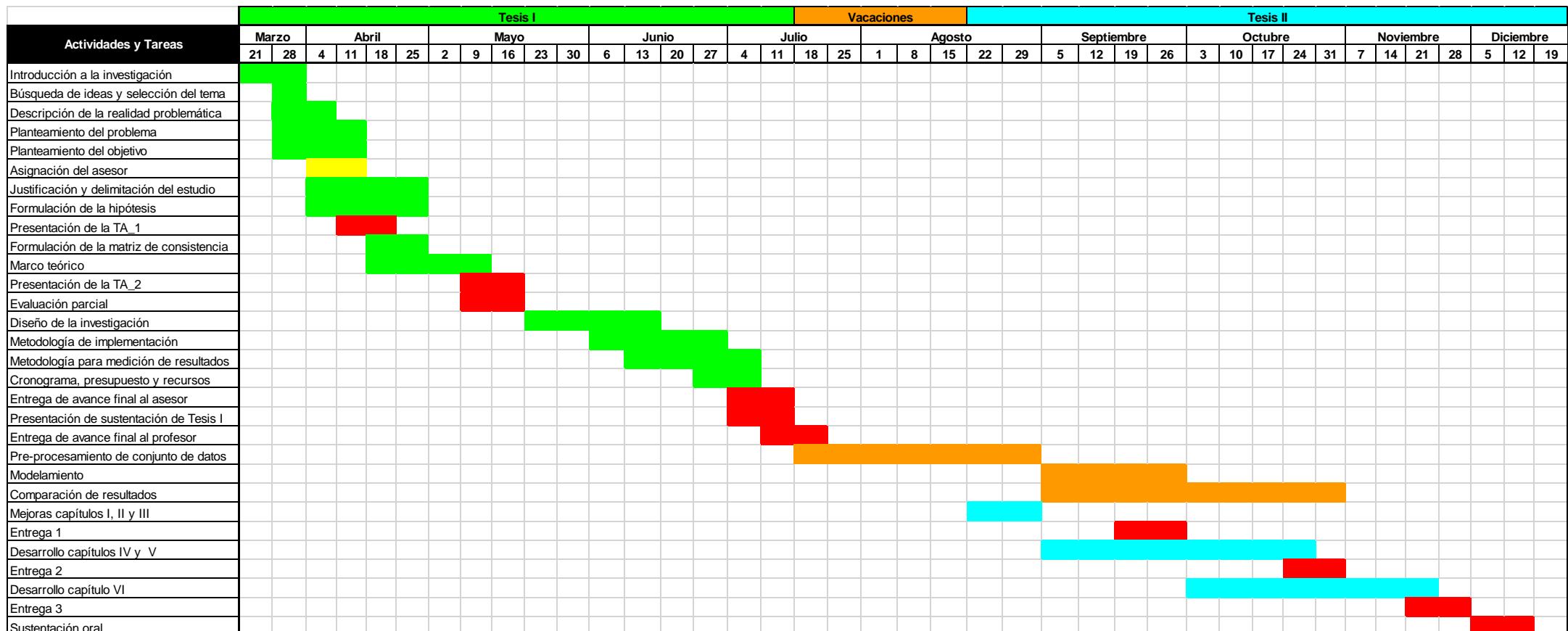
## **Despliegue**

Luego de explotar al máximo la utilidad de los modelos, se procede a unir los aquellos con mejor desempeño y resultados dados para que puedan ser implementados dentro de la organización. Asimismo, se les deben integrar tareas una vez implementados para la toma de decisiones organizacionales que ayuden a conseguir los objetivos trazados.

Para ello, se debe comenzar con planificar la fase de despliegue, monitorización y mantenimiento de acuerdo al caso. Al entrar en operación el o los modelos de Minería de Datos, se debe generar un informe final que contenga los procesos seguidos y resultados obtenidos. Finalmente, como parte final del proyecto, este debe pasar a la fase de revisión para confirmar la correcta implementación.

### **3.7. Cronograma de actividades y presupuesto**

Se elaboró un cronograma de actividades de todo el proyecto mostrado en la **Figura 49**, comenzando con la primera parte del curso de Tesis desde el mes de marzo hasta la finalización de la segunda parte del mismo y sustentación respectiva en el mes de diciembre.



**Figura 49.** Cronograma de actividades del proyecto solución.

**Fuente:** Elaboración propia.

La partida presupuestal de todo el proyecto se divide en dos partes: los costos personales del autor y los costos de las herramientas para la operación del proyecto.

Los costos personales del autor del trabajo de tesis se muestran en la **Tabla 8**. Estos incluyen los recursos utilizados durante el primer semestre del año 2019, así como la inclusión de la herramienta tecnológica (laptop) para la segunda mitad del año.

Item	Tiempo (horas)	Costo (soles)	Subtotal
<b>Recursos materiales</b>			
Laptop i3 (para Tesis I)		S/.1,500.00	S/.1,500.00
Laptop i7 (para Tesis II)		S/.4,500.00	S/.4,500.00
<b>Recursos humanos</b>			
Avance de tesis	310	S/.30.00	S/.9,300.00
Movilidad a clases	30	S/.40.00	S/.1,200.00
<b>Servicios generales</b>			
Internet + luz	9	S/.80.00	S/.720.00
<b>Total</b>			<b>S/.17,220.00</b>

**Tabla 8.** Presupuesto de los costos personales del autor.

**Fuente:** Elaboración propia.

Si el valor total se evalúa en dólares americanos (USD), el monto total asignado sería un aproximado de \$ 5,156.

Además de la partida presupuestal del proyecto, se realizó un benchmarking de precios de diferentes empresas que ofrecen servidores y computadoras en la nube. Entre ellas se encuentran Alibaba, Amazon Web Service (AWS), Paperspace, Google Cloud, Vast y Azure. Debido a que los precios referenciales son del territorio de los Estados Unidos de América, la divisa será en dólares americanos (USD). Para conocer los precios en la moneda local, es decir, en soles peruanos (PEN) se tendrá que realizar la conversión respectiva con una tasa de cambio promedio proyectada al cierre del año de 3.34 (Scotiabank, 2019).

		Alibaba			AWS		
		ecs.sn2ne.large	ecs.sn2ne.2xlarge	ecs.i2.2xlarge	m5.large	m5.2xlarge	i3.2xlarge
vCPU's		2	8	8	2	8	8
Memoria GB		8	32	64	8	32	61
Disco local				1*1788 GiB NVMe SSD			1x1900 NVMe SSD
<b>Linux pay-as-you-go w/IP pública</b>							
US West (NoCal)	Mensual	\$64.000	\$249.760	\$329.400	\$81.760	\$327.040	\$502.240
	Pay-As-You-Go	\$0.124	\$0.486	\$0.656	\$0.112	\$0.448	\$0.688
Singapur	Mensual	\$60.990	\$237.880	N/A	\$87.600	\$350.400	\$546.040
	Pay-As-You-Go	\$0.133	\$0.519	\$0.712	\$0.120	\$0.480	\$0.748
Frankfurt	Mensual	\$64.180	\$250.620	\$351.600	\$83.950	\$335.800	\$543.120
	Pay-As-You-Go	\$0.127	\$0.498	\$0.708	\$0.115	\$0.460	\$0.744
China	Mensual	\$46.780	\$177.890	\$323.900	-	-	-
	Pay-As-You-Go	\$0.158	\$0.577	\$1.073	-	-	-
<b>Windows Server 2008 R2 Enterprise Mensual w/IP pública</b>							
US West (NoCal)	Mensual	\$136.410	\$539.450	\$584.600	\$148.920	\$595.680	\$770.880
	Pay-As-You-Go	\$0.220	\$0.868	\$1.004	\$0.204	\$0.816	\$1.056
Singapur	Mensual	\$130.810	\$517.120	N/A	\$154.760	\$619.040	\$814.680
	Pay-As-You-Go	\$0.228	\$0.901	\$1.064	\$0.212	\$0.848	\$1.116
Frankfurt	Mensual	\$134.000	\$529.880	\$606.800	\$151.110	\$604.440	\$811.760
	Pay-As-You-Go	\$0.223	\$0.880	\$1.060	\$0.207	\$0.828	\$1.112
China	Mensual	\$46.780	\$177.890	\$323.900	-	-	-
	Pay-As-You-Go	\$0.158	\$0.577	\$1.073	-	-	-
<b>Transferencia de datos desde la Nube a Internet, por GB (AWS es escalonada, precio es por primeros 10 TB)</b>							
US West (NoCal)		\$0.077	Mensual	\$2.387	\$0.090	Mensual	\$2.790
Singapur		\$0.081	Mensual	\$2.511	\$0.120	Mensual	\$3.720
Frankfurt		\$0.070	Mensual	\$2.170	\$0.090	Mensual	\$2.790
China		\$0.123	Mensual	\$3.813	N/A	Mensual	-

**Tabla 9.** Cuadro comparativo entre precios de servidores en la nube de Alibaba y Amazon Web Service (AWS).

**Fuente:** (Chapel, 2018).

	<b>GPU+ / P4000</b>	<b>P5000</b>	<b>P6000</b>	<b>V100</b>
RAM (GB)	30	30	30	30
CPU	8	8	8	8
GB GPU	8	16	24	16
CUDA cores	1,792	2,560	3,840	-
TeraFLOPs	-	-	-	112
Precio (x hora)	\$0.51	\$0.78	\$1.10	\$2.30
Precio (x mes)	\$15.81	\$24.18	\$34.10	\$71.30

**Tabla 10.** Descripción de características y precios de GPU dedicada en Paperspace.**Fuente:** (Paperspace, 2018).

	Instancias CPU			Instancias GPU				TPU
	<b>G1</b>	<b>G6</b>	<b>G12</b>	<b>K80</b>	<b>P100</b>	<b>V100</b>	<b>V100 x8</b>	<b>TPUv2</b>
RAM (GB)	1.7	24.0	64.0	12.0	24.0	30.0	130.0	16.0
vCPU	1	6	12	2	4	8	20	-
GDDRS (GB)	-	-	-	12	16	16	128	-
memory bandwidth (GB/s)	-	-	-	480	732	900	7,200	2,400
CUDA cores	-	-	-	2,496	3,584	5,120	40,960	-
TeraFLOPs	-	-	-	-	-	-	-	180
Precio (x hora)	\$0.042	\$0.120	\$0.230	\$0.250	\$0.590	\$1.150	\$8.430	\$8.420
Precio (x mes)	\$1.296	\$3.720	\$7.130	\$7.750	\$18.290	\$35.650	\$261.330	\$261.020

**Tabla 11.** Cuadro comparativo entre instancias CPU, GPU y TPU en Paperspace.**Fuente:** (Paperspace, 2018).

Cloud Service	NVIDIA GPU	CPUs	GPU RAM	CPU RAM	Costo por Hora	Wall Time	Costo por Train	Costo Mensual
Google Colab	K80	1	12	13	\$0.000	31.17	\$0.000	\$0.000
Google Cloud Compute Engine	P100	6	16	20	\$0.500	5.32	\$0.044	\$15.500
Google Cloud Compute Engine	K80	6	12	17	\$0.200	18.13	\$0.060	\$6.200
Google Cloud Compute Engine	V100	8	16	20	\$0.820	3.83	\$0.052	\$25.420
Google Cloud Compute Engine	P4	4	8	26	\$0.330	10.28	\$0.057	\$10.230
Google Cloud Compute Engine	V100 x 2	8	32	30	\$1.570	3.63	\$0.095	\$48.670
Google Cloud Compute Engine	V100 x 4	8	64	30	\$3.050	3.38	\$0.172	\$94.550
AWS EC2	K80 (p2.xlarge)	4	12	61	\$0.280	20.9	\$0.098	\$8.680
AWS EC2	K80 x 8 (p2.8xlarge)	32	96	488	\$2.350	16.12	\$0.631	\$72.850
AWS EC2	V100 (p3.2xlarge)	8	16	61	\$1.050	3.85	\$0.067	\$32.550
AWS EC2	V100 x 4 (p3.8xlarge)	64	128	488	\$4.050	2.97	\$0.200	\$125.550
vast.ai	GTX 1070 Ti	4	8.1	16	\$0.060	7.23	\$0.008	\$1.860
Paperspace	Quadro M4000	8	8	30	\$0.510	8.3	\$0.071	\$15.810

**Tabla 12.** Cuadro comparativo de diferentes servicios en la nube (Google Cloud, AWS, vast.ai, Paperspace).**Fuente:** (Hale, 2018).

Instancia	Núcleos	RAM (GB)	Almacenamiento temporal (GB y \$/h)
A0	1	0.75	20.00 0.02
A1	1	1.75	225.00 0.08
A2	2	3.50	490.00 0.16
A3	4	7.00	1,000.00 0.32
A4	8	14.00	2,040.00 0.64
A5	2	14.00	490.00 0.35
A6	4	28.00	1,000.00 0.71
A7	8	56.00	2,040.00 1.41

**Tabla 13.** Comparación de precios de distintas instancias de Azure.**Fuente:** (Microsoft Azure).

## 4. CAPÍTULO IV: DESARROLLO DEL EXPERIMENTO

### 4.1. Construcción de los conjuntos finales de datos

El conjunto total de datos que se utilizó para el experimento (siguiendo el paper del décimo antecedente) fue separado en tres partes: metainformación, contenido visual y contenido textual.

#### 4.1.1. Metainformación

Como se mencionó en las Técnicas de recolección de datos en el Capítulo III, la data original fue obtenida de la página Web Robots, sitio web encargado de descargar información web por scraping, accediendo al enlace para descargar conjuntos de datos de Kickstarter (<https://webrOBots.io/kickstarter-datasets/>). Se descargaron archivos de valores separados por comas (.csv) particionados, comprimidos y agrupados por mes, scrapeados entre los periodos de noviembre del 2015 y agosto del 2019, como se aprecia en la **Figura 50** (Web Robots). De acuerdo con la información de los creadores del sitio Web Robots, se ejecutan robots en dos servidores en la nube encargados de recolectar en un determinado punto del día y una vez al mes información de las campañas que aparecen en Kickstarter (Web Robots).

The screenshot shows a page titled "Kickstarter Datasets". It contains a paragraph explaining that a scraper robot crawls all Kickstarter projects and collects data in CSV and JSON formats monthly from March 2016. Below this, there are two sections: "2019" and "2018", each listing specific dates with links to download the data in both JSON and CSV formats.

Year	Date	Format
2019	2019-08-15	[JSON] – [CSV]
	2019-07-13	[JSON] – [CSV]
	2019-06-13	[JSON] – [CSV]
	2019-05-16	[JSON] – [CSV]
	2019-04-18	[JSON] – [CSV]
	2019-03-14	[JSON] – [CSV]
	2019-02-14	[JSON] – [CSV]
	2019-01-17	[JSON] – [CSV]
2018	2018-12-13	[JSON] – [CSV]
	2018-11-15	[JSON] – [CSV]
	2018-10-18	[JSON] – [CSV]
	2018-09-13	[JSON] – [CSV]
	2018-08-16	[JSON] – [CSV]
	2018-07-12	[JSON] – [CSV]
	2018-06-14	[JSON] – [CSV]
	2018-05-17	[JSON] – [CSV]
	2018-04-12	[JSON] – [CSV]
	2018-03-15	[JSON] – [CSV]
	2018-02-15	[JSON] – [CSV]
	2018-01-12	[JSON] – [CSV]

**Figura 50.** Vista de la página de data disponible recolectada de Kickstarter.

**Fuente:** (Web Robots).

Cada archivo descargado por periodo mensual es descomprimido, y las particiones en formato .csv que contienen son juntadas mediante el siguiente proceso:

- ✓ Descargar librerías correspondientes de Python (pandas, numpy, glob, math).

- ✓ Dirigirse a la ruta de destino.
- ✓ Crear carpetas por mes para almacenar archivos particionados.
- ✓ Ejecutar algoritmo para unir archivos particionados csv dentro de una misma carpeta.

Cuando el algoritmo finaliza su ejecución, se creará un nuevo archivo separado por comas en cada carpeta por mes, cuyo tamaño oscila entre 1 y 5 gigabytes (GB) en total por cada uno. Con el fin de ahorrar espacio en memoria dentro de la computadora, los archivos particionados son eliminados y solamente quedan los archivos nuevos generados.

En la **Figura 51** se aprecia como ejemplo el resumen de uno de los datasets obtenidos.

```
In [7]: data_combinada_201907.shape    ##Originalmente habían 212,378 registros de todos los proyectos
Out[7]: (212378, 37)

In [8]: data_combinada_201907.columns
Out[8]: Index(['backers_count', 'blurb', 'category', 'converted_pledged_amount',
       'country', 'created_at', 'creator', 'currency', 'currency_symbol',
       'currency_trailing_code', 'current_currency', 'deadline',
       'disable_communication', 'friends', 'fx_rate', 'goal', 'id',
       'is_backing', 'is_starrable', 'is_starred', 'launched_at', 'location',
       'name', 'permissions', 'photo', 'pledged', 'profile', 'slug',
       'source_url', 'spotlight', 'staff_pick', 'state', 'state_changed_at',
       'static_usd_rate', 'urls', 'usd_pledged', 'usd_type'],
      dtype='object')
```

**Figura 51.** Tamaño y columnas del conjunto de datos, periodo de julio del 2019.

**Fuente:** Elaboración propia.

A continuación, estos nuevos conjuntos de datos generados son filtrados para que contengan solamente proyectos tecnológicos. Al no existir la variable *main\_category* (la cual clasifica a un proyecto dentro de las categorías mostradas en la **Figura 2**) dentro de los datasets obtenidos, se usó la variable *source\_url* para seleccionar solamente aquellos registros que contengan la cadena de caracteres “<https://www.kickstarter.com/discover/categories/technology>”.

```
In [40]: df_201907_filtrada.shape
Out[40]: (21398, 37)
```

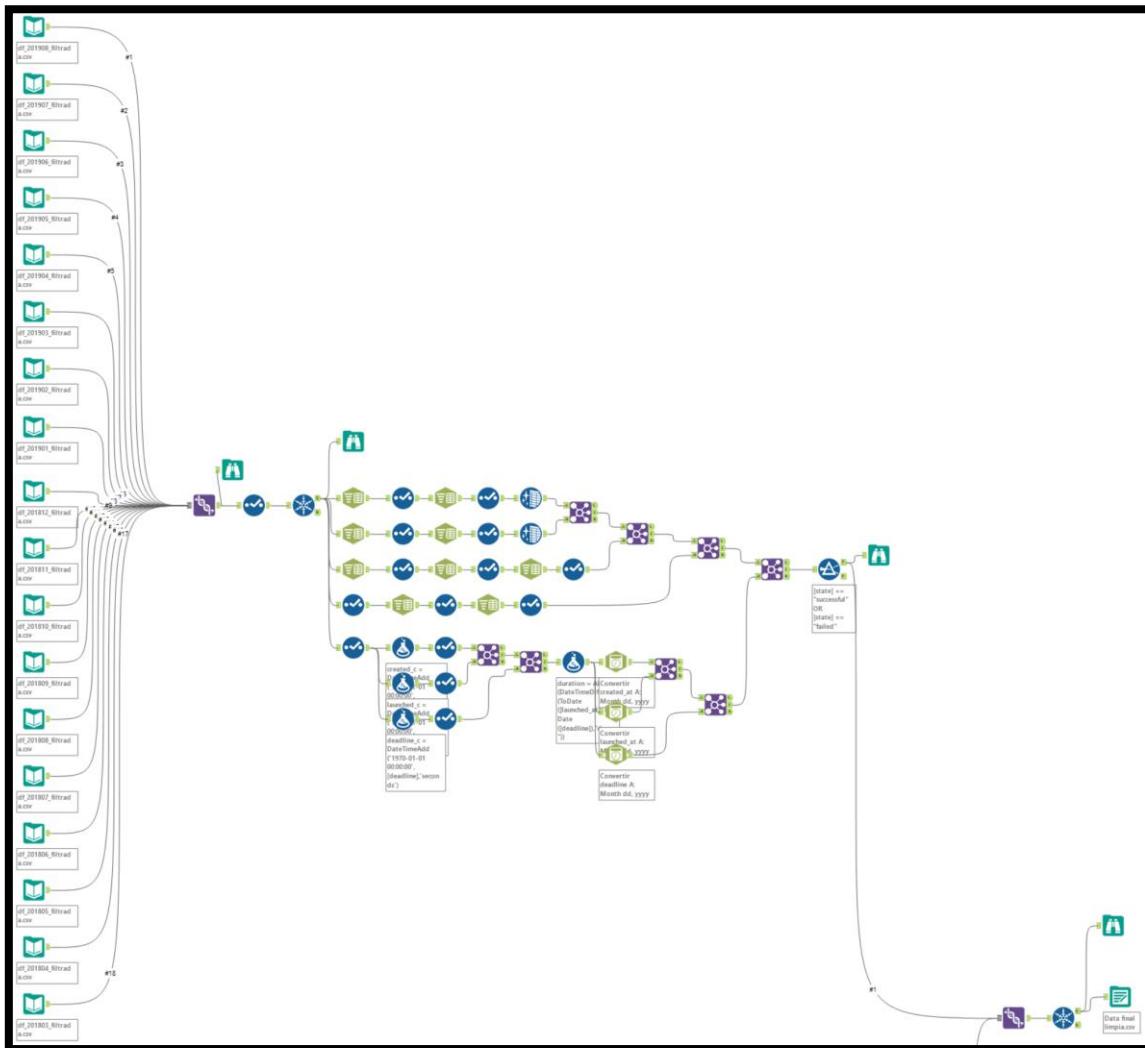
**Figura 52.** Conjunto de datos filtrado por categoría, periodo de julio del 2019.

**Fuente:** Elaboración propia.

Cuando se repitió este procedimiento con cada conjunto de datos, se notó que la proporción de proyectos tecnológicos en Kickstarter respecto al total de proyectos representa aproximadamente el 10%. Esto se infirió al comparar los tamaños (cantidad de registros por fila) del conjunto de datos inicial y del filtrado.

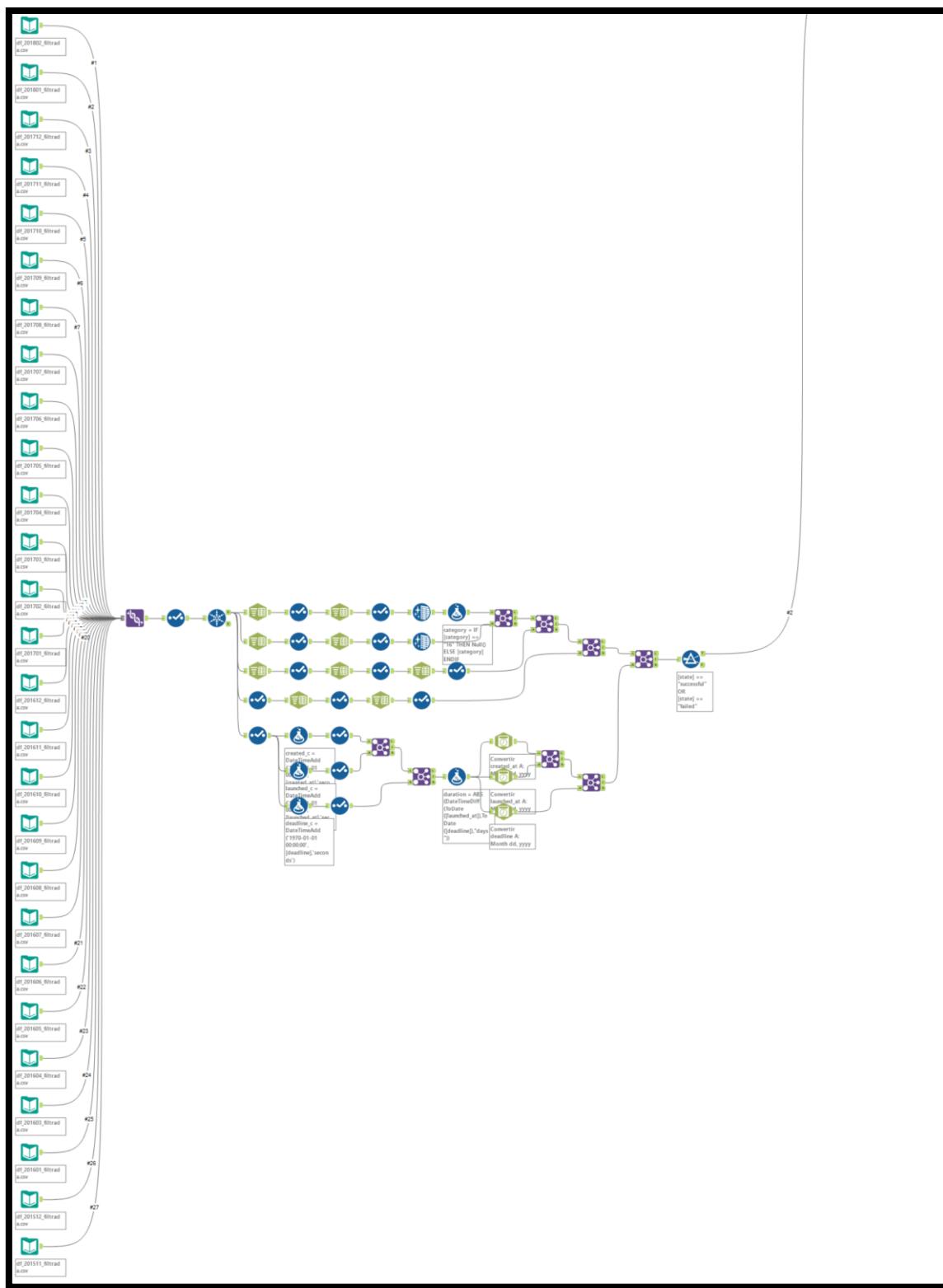
La siguiente fase del pre-procesamiento fue la de seleccionar las variables de los nuevos conjuntos de datos filtrados que se usarán para generar el dataset final. Para ello, se usó el software Alteryx Designer, el cual permite desarrollar flujos de trabajo para preparar, unir y analizar volúmenes de datos complejos de distintas fuentes.

Con los conjuntos de datos filtrados se creó el flujo de trabajo representado en las **Figura 53** y **Figura 54**.



**Figura 53.** Parte superior del flujo de trabajo de la data final.

**Fuente:** Elaboración propia.



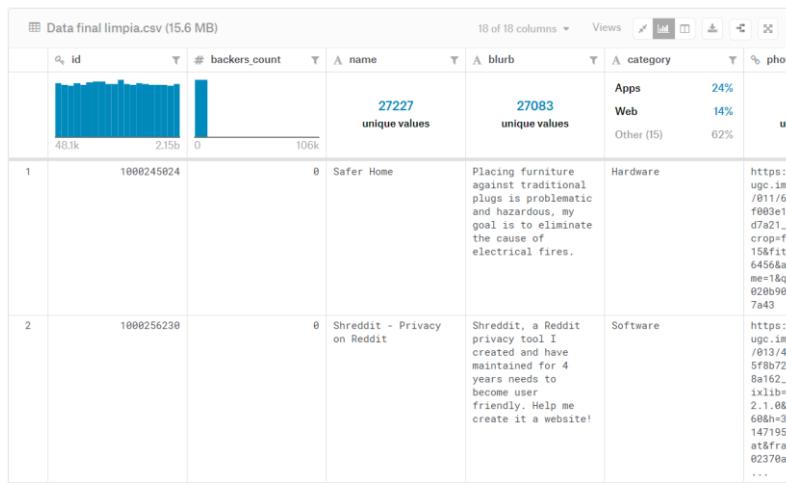
**Figura 54.** Parte inferior del flujo de trabajo de la data final.

**Fuente:** Elaboración propia.

El flujo comienza con la carga de los datos de entrada, los cuales son los 45 archivos separados por coma por cada periodo mensual desde noviembre del 2015 hasta septiembre del 2019. Luego, se realizaron dos uniones en vez de una sola, esto debido a que, a partir de marzo del 2018, algunas de las variables y valores de estas son diferentes a la de sus predecesoras. Sin embargo, esto no afectará al proceso más adelante ya que ambas uniones serán juntadas por las mismas variables.

En cada una de las dos uniones, se seleccionaron las variables de la Tabla N° 3, se realizó limpieza de datos para las variables *category*, *location*, *photo* y *urls*, y se transformaron las variables numéricas en milisegundos *created\_at*, *launched\_at* y *deadline* a variables de fecha. Esto último permitió crear la variable *duration* para determinar la duración de la campaña de un proyecto obtenida de la diferencia entre la fecha de culminación (*deadline*) y la fecha de lanzamiento (*launched\_at*). Asimismo, culminadas estas labores y unidas las variables limpias, se filtró en la variable *state* solo aquellos registros que contengan el valor de “successful” o “failed” ya que se analizarán solamente los proyectos que han sido exitosos o fracasados. Por último, la variable *urls* contiene los enlaces directos de los proyectos. Esta variable sirvió para aplicar scraping a estos sitios web y obtener la descripción de cada proyecto con el fin de realizar Minería de texto más adelante.

El flujograma culmina con la generación de un archivo de valores separados por coma (.csv) guardado en memoria local. Para poder trabajar en Colaboratory de Google Drive, el archivo generado fue subido a la plataforma Kaggle de manera pública para que pueda ser descargada a través del API de la web.



**Figura 55.** Visualización del archivo de metainformación subido a Kaggle.

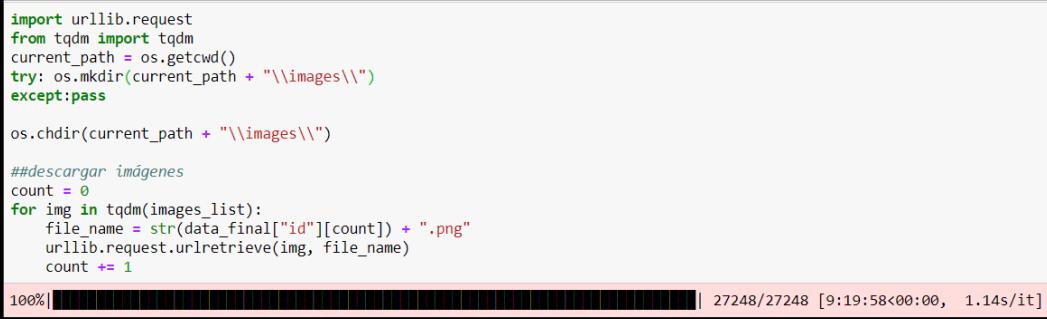
**Fuente:** Elaboración propia.

#### 4.1.2. Contenido visual

Una vez obtenido el conjunto de datos final de la metainformación, se considerará la variable *photo* para descargar todas las imágenes de los proyectos que se encuentran en Kickstarter. Esta contiene los enlaces URL de las imágenes de cada proyecto.

Para ello, se creó un algoritmo, usando las librerías de Python **urllib.request** y **tqdm**, que descargó cada una de ellas y las almacenó en una carpeta de la memoria local llamada “images”, asignándole el id del proyecto al que pertenece como nombre, así como extensión “.png”.

El tiempo total de descarga de las 27,248 imágenes fue de 9 horas y 20 minutos aprox. como se aprecia en la **Figura 56**.



```

import urllib.request
from tqdm import tqdm
current_path = os.getcwd()
try: os.mkdir(current_path + "\\images\\")
except:pass

os.chdir(current_path + "\\images\\")

##descargar imágenes
count = 0
for img in tqdm(images_list):
    file_name = str(data_final["id"][count]) + ".png"
    urllib.request.urlretrieve(img, file_name)
    count += 1

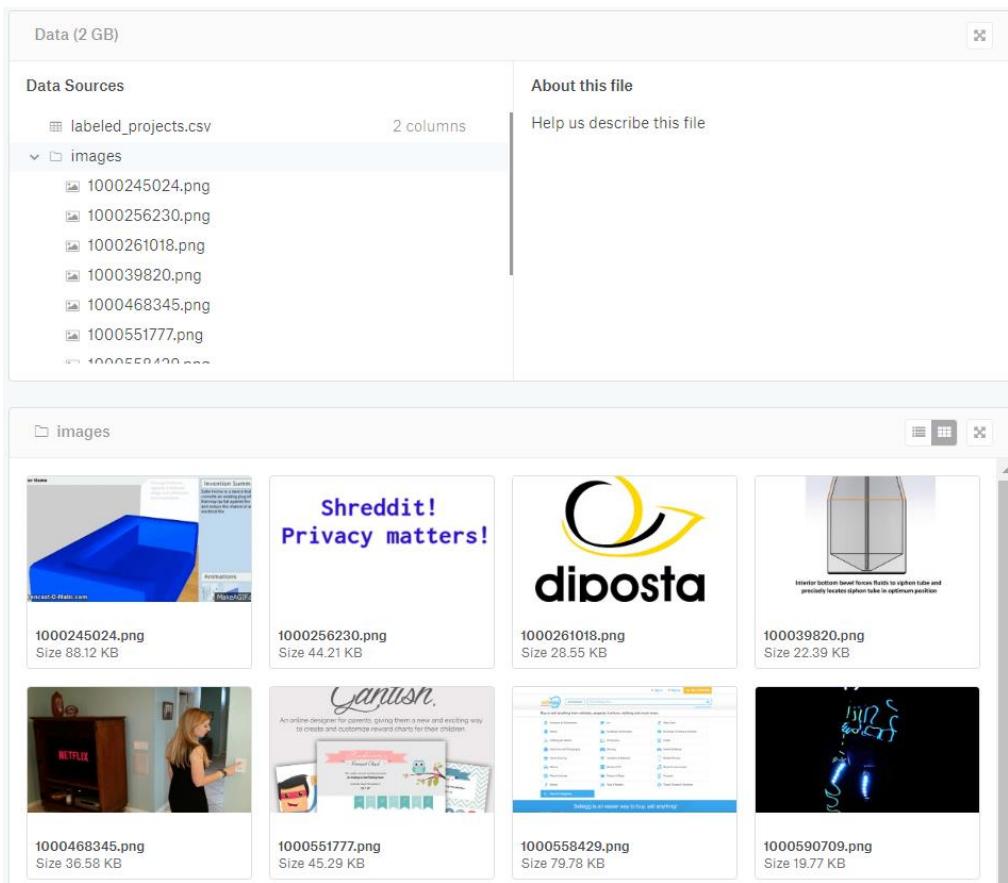
```

The screenshot shows a Jupyter Notebook cell with Python code for downloading images. The code uses the `urllib.request` library and the `tqdm` progress bar module. It creates a directory named 'images' in the current working directory, changes to it, and then loops through a list of URLs ('images\_list') to download each one as a file named after its ID (e.g., '1.png', '2.png', etc.). A progress bar at the bottom indicates the completion of 27,248 files in 9 hours and 1.14 seconds per iteration.

**Figura 56.** Proceso de descarga de imágenes.

**Fuente:** Elaboración propia.

Para poder trabajar en Colaboratory de Google Drive, la carpeta de imágenes fue comprimida junto a un archivo .csv que contiene dos columnas: el nombre del proyecto (id) y la etiqueta del estado (state) transformando los valores “successful” por 1 y “failed” por 0. Este nuevo archivo fue subido a la plataforma Kaggle de manera pública para que pueda ser descargada a través del API de la web.



**Figura 57.** Visualización del archivo comprimido de imágenes en Kaggle.

**Fuente:** Elaboración propia.

#### 4.1.3. Contenido textual

Finalmente, el contenido textual, que consiste en las descripciones de cada proyecto visible en Kickstarter, fue descargado gracias a la variable *urls*. Esta contiene las direcciones URLs de los proyectos en Kickstarter.

Para ello, y como se describe en la **Figura 58**, se elaboró un algoritmo que, mediante la navegación al contenido de estas páginas a través de un agente falso, se dirigió a las descripciones de los proyectos identificando las etiquetas con clase llamada “**rte\_content js-full-description responsive-media**” y las almacenó en un vector vacío, uniendo previamente los párrafos y eliminando caracteres especiales, para posteriormente unirlo a la variable *id* (que permite identificar qué descripción le corresponde a su proyecto) y guardarlo en un archivo de valores separados por coma (.csv). En caso el algoritmo no encuentre esta clase dentro de las páginas (*IndexError*), el vector almacenaba un registro nulo.

```

def getPageText(url):
    # Crear agente falso para scrapear
    ua = UserAgent()
    user_agent = ua.chrome
    # Solicitar uso de librería urllib con agente falso
    request = urllib.request.Request(url, headers = {"User-Agent":user_agent})
    # Obtener contenido de urls mediante Librería urllib
    data = urllib.request.urlopen(request)
    # parse as html structured document
    bs = BeautifulSoup(data, "html.parser")
    # Buscar todos los tags div con una determinada clase
    # A partir de acá, se usa un "try" y un "except" porque hay páginas que no muestran su contenido
    # Para evitar que salga el mensaje de error, se llenarán como null los contenidos que no se puedan descargar
    # El try contiene el algoritmo para descargar la descripción de un proyecto
    try:
        description = bs.find_all("div", {"class":"rte_content js-full-description responsive-media"})
        # Encontrar todos los párrafos
        description = description[0].find_all("p")
        # Crear array vacío donde se almacenará el contenido descargado
        project_description = []
        # Iteración para agregando en lista cada descripción descargada
        for link in description:
            project_description.append(link.text)
        # Junta párrafos separados en un solo vector, separándolos con un espacio
        project_description = ' '.join(project_description)
        # Eliminar caracteres especiales
        project_description = project_description.replace(u'\xa0', u' ')
        project_description = project_description.replace(u'\n', u' ')
        # Y el except sirve para llenar valores nulos en el array en caso de error
    except (IndexError, ValueError):
        project_description = 'null'

    # Eliminar saltos de línea
    return Newlines.sub('\n', project_description)

```

**Figura 58.** Función del algoritmo web scraping de la descripción de proyectos.

**Fuente:** Elaboración propia.

Debido a la gran cantidad de memoria y tiempo que iba a presentar este proceso, se determinó fraccionar los 27,251 proyectos en tres partes y repetir el mismo en cada uno de ellos, como se aprecia en las **Figura 59** y **Figura 60**. El tiempo aproximado de descarga de cada fracción fue de 6 horas.

```

# Asignar urls de búsqueda (origen)
urls_primera_parte = urls_list[0:9083]
urls_segunda_parte = urls_list[9084:18168]
urls_tercera_parte = urls_list[18169:27251]
#urls

ids_primera_parte = data_final["id"][0:9083]
ids_segunda_parte = data_final["id"][9084:18168]
ids_tercera_parte = data_final["id"][18169:27251]

```

**Figura 59.** Fraccionamiento de la data total en tres partes.

**Fuente:** Elaboración propia.

```
# Hacemos lo mismo con la segunda parte
description_txt = [getPageText(url) for url in urls_segunda_parte]
df = {"id":ids_segunda_parte, "description":description_txt}
data_final = pd.DataFrame(df)
export_csv = data_final.to_csv (r'D:\TTT\DB\dataset_descripciones\data_final_parte2.csv', index = None, header=True)

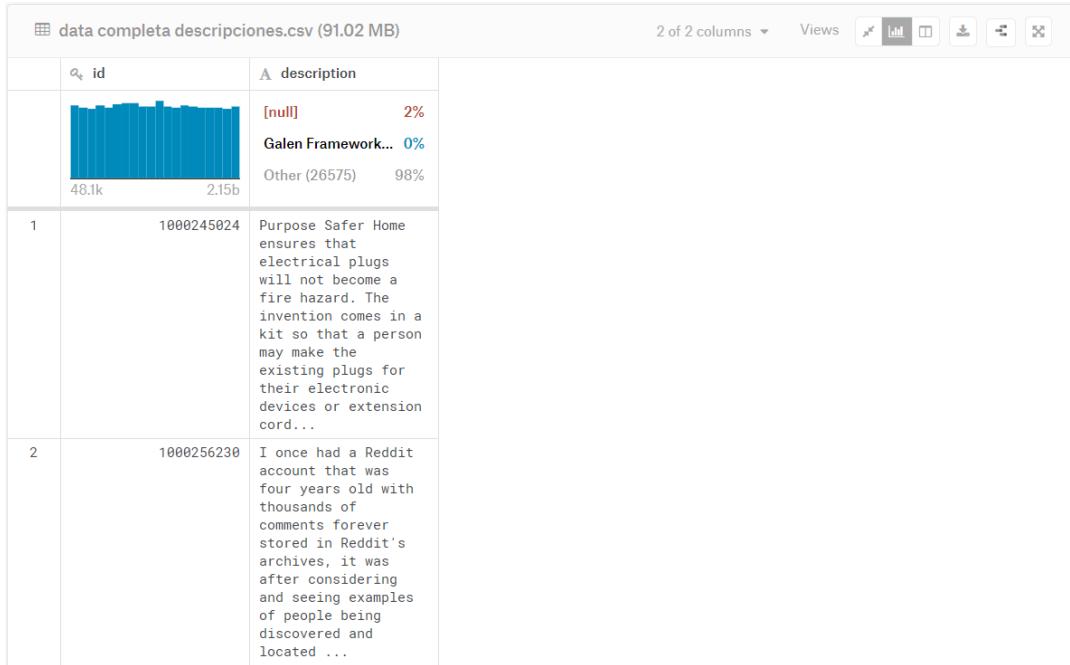
# Hacemos lo mismo con la tercera parte
description_txt = [getPageText(url) for url in urls_tercera_parte]
df = {"id":ids_tercera_parte, "description":description_txt}
data_final = pd.DataFrame(df)
export_csv = data_final.to_csv (r'D:\TTT\DB\dataset_descripciones\data_final_parte3.csv', index = None, header=True)
```

**Figura 60.** Repetición del proceso para las fracciones restantes.

**Fuente:** Elaboración propia.

Finalmente, las tres partes fueron unidas, se reemplazaron los valores nulos por espacios en blanco y se guardó como un nuevo archivo de valores separados por coma (.csv) en código Unicode UTF-8 para la lectura de caracteres no alfabéticos.

Para poder trabajar en Colaboratory de Google Drive, el archivo generado fue subido a la plataforma Kaggle de manera pública para que pueda ser descargada a través del API de la web.



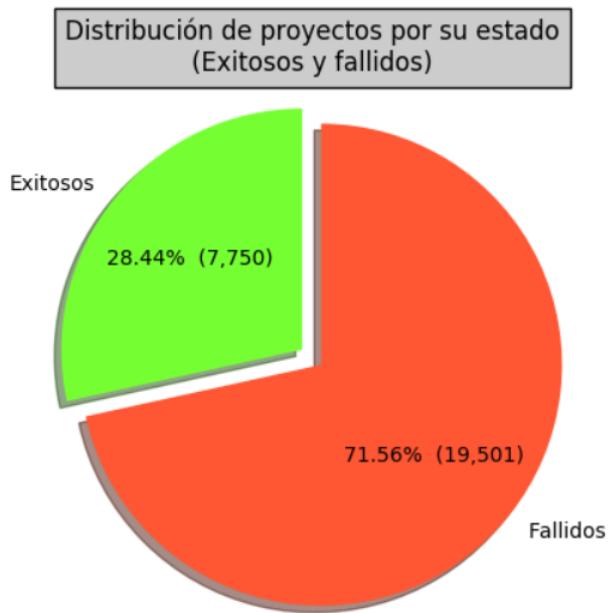
**Figura 61.** Visualización del archivo de descripciones subido a Kaggle.

**Fuente:** Elaboración propia.

## 4.2. Análisis exploratorio de los datos

### 4.2.1. Metainformación

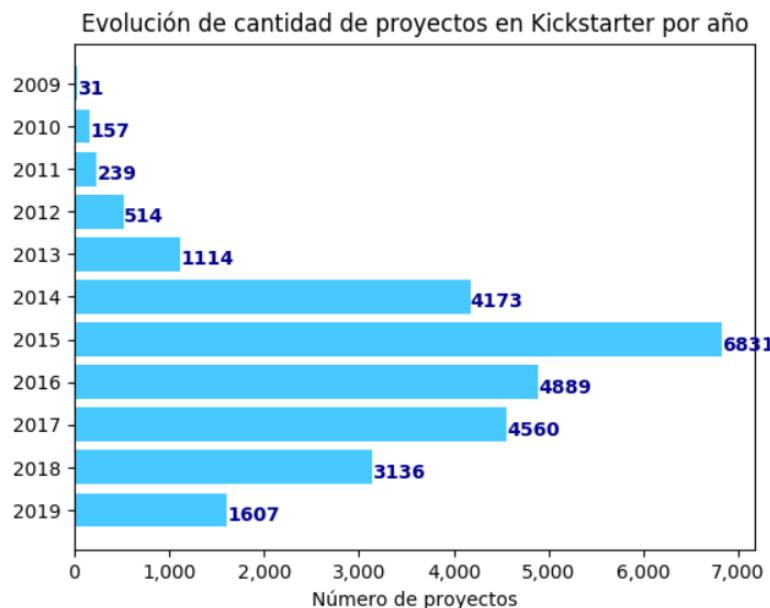
Comenzando con la variable dependiente (objetivo) *state* se distribuyen sus registros como se representa en la **Figura 62**.



**Figura 62.** Distribución total de proyectos tecnológicos por su estado.

**Fuente:** Elaboración propia.

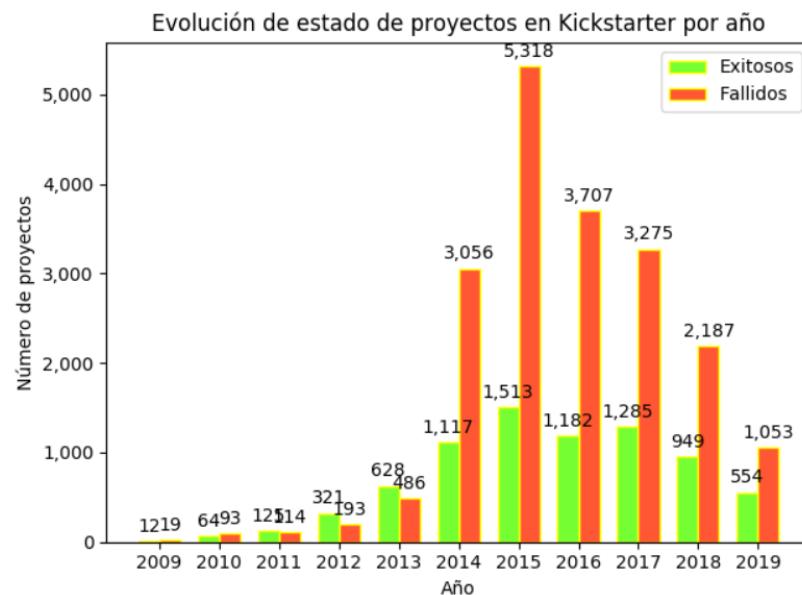
Asimismo, la **Figura 63** muestra la distribución de los proyectos tecnológicos en Kickstarter por año.



**Figura 63.** Evolución de cantidad de proyectos tecnológicos por año.

**Fuente:** Elaboración propia.

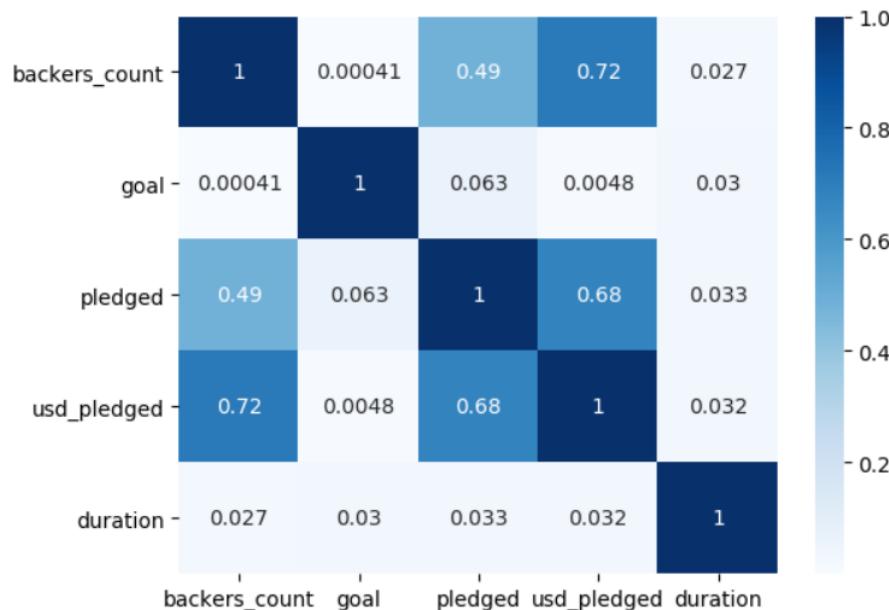
Finalmente, en la **Figura 64** se observa la distribución de los proyectados tecnológicos en Kickstarter agrupados por su estado.



**Figura 64.** Evolución de proyectos tecnológicos, por su estado y año.

**Fuente:** Elaboración propia.

Analizando ahora las variables independientes numéricas mencionadas en la Tabla N° 6, se realizó la correlación entre ellas para ver si se cumple el supuesto de no-colinealidad, es decir, la no existencia de colinealidad perfecta o parcial (correlaciones muy altas) entre las variables independientes. Para ello, se representan el nivel de sus correlaciones entre sí en la **Figura 65**.

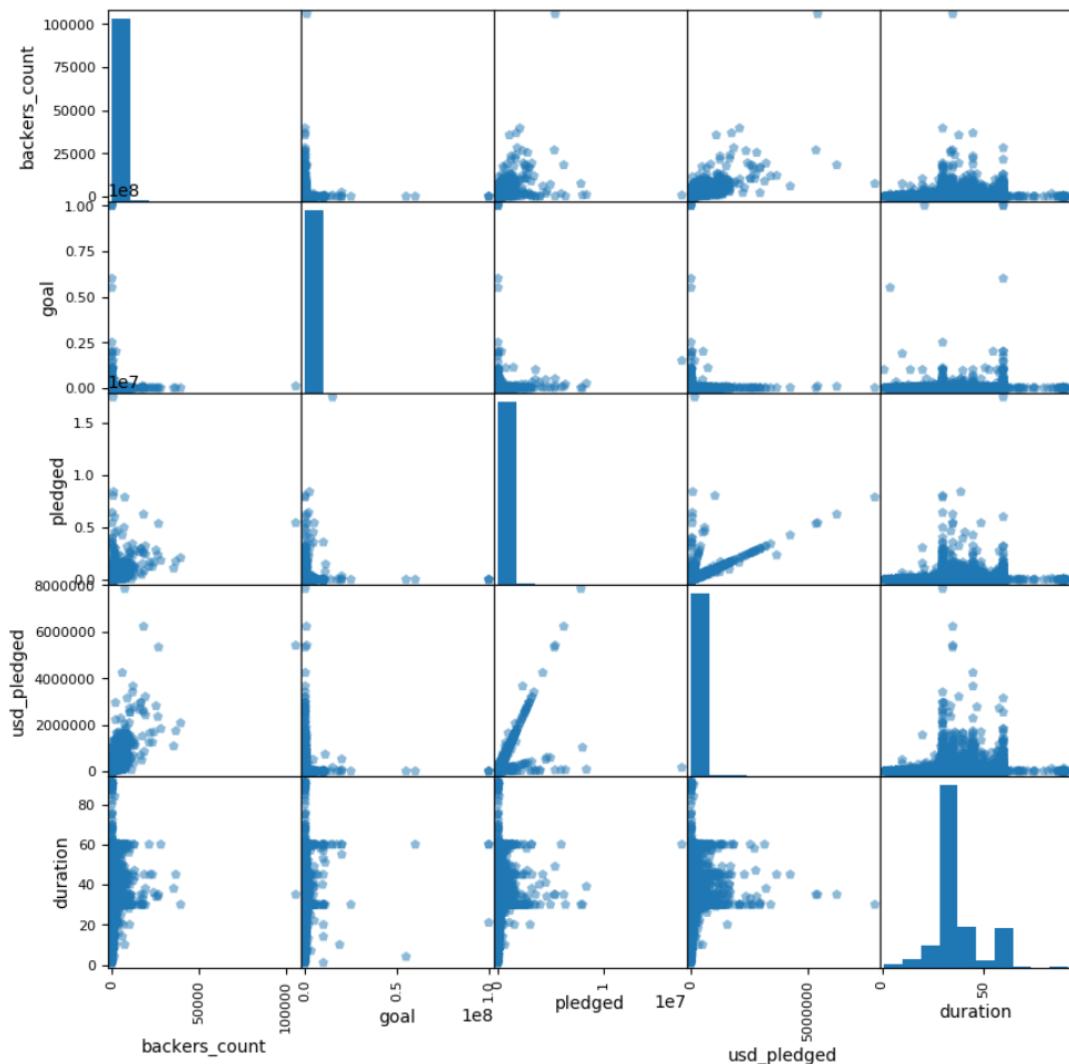


**Figura 65.** Matriz de correlaciones entre variables independientes.

**Fuente:** Elaboración propia.

Como se puede apreciar en la matriz, la variable *usd\_pledged* está altamente correlacionada con las variables *backers\_count* y *pledged* (casi un 70% en ambas). Esto quiere decir que esta variable no es significativa para el presente trabajo porque, de ser tomada en cuenta, explicaría muy similar que las otras dos comentadas y reduciría la precisión del modelo.

Asimismo, si se observan los registros desde una matriz que contiene, además de gráficos de dispersión de las correlaciones, histogramas de las variables independientes como en la **Figura 66**, se confirma y concluye no utilizarla para los experimentos.



**Figura 66.** Matriz de correlaciones e histogramas de variables independientes.

**Fuente:** Elaboración propia.

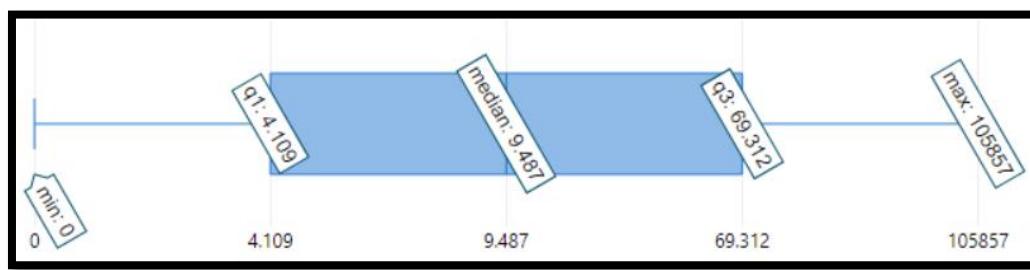
En adición, de todas ellas, la variable *duration* es la única que sigue una distribución normal a diferencia del resto. Estas últimas fueron normalizadas luego.

Finalmente, después del análisis anterior, se determinó usar las siguientes variables:

- Cantidad de patrocinadores (*backers\_count*):

Los principales datos estadísticos de esta variable son los siguientes:

- Rango de valores: [0; 105,857]
- Media: 208.710469340575
- Mediana: 9.487
- Desviación estándar: 1,179.68237749203
- Varianza: 1,391,650.51176525



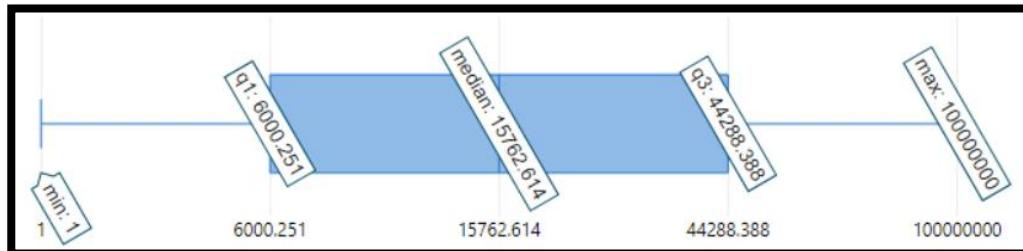
**Figura 67.** Caja de bigotes de la variable *backers\_count*.

**Fuente:** Elaboración propia.

- Monto meta (*goal*):

Los principales datos estadísticos de esta variable son los siguientes:

- Rango de valores: [1; 100,000,000]
- Media: 91,263.9666162825
- Mediana: 15,762.614
- Desviación estándar: 1,259,282.1587922
- Varianza: 1,585,791,555,452.35



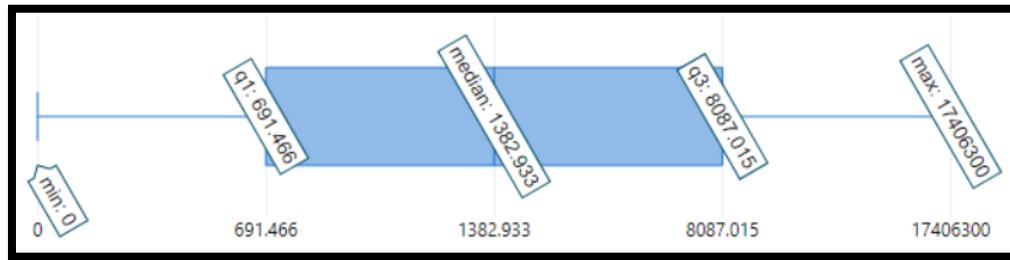
**Figura 68.** Caja de bigotes de la variable *goal*.

**Fuente:** Elaboración propia.

- Monto patrocinado o invertido (*pledged*):

Los principales datos estadísticos de esta variable son los siguientes:

- Rango de valores: [0; 17,406,300]
- Media: 34,668.5134710787
- Mediana: 1,382.933
- Desviación estándar: 226,763.900313481
- Varianza: 51,421,866,485.3822



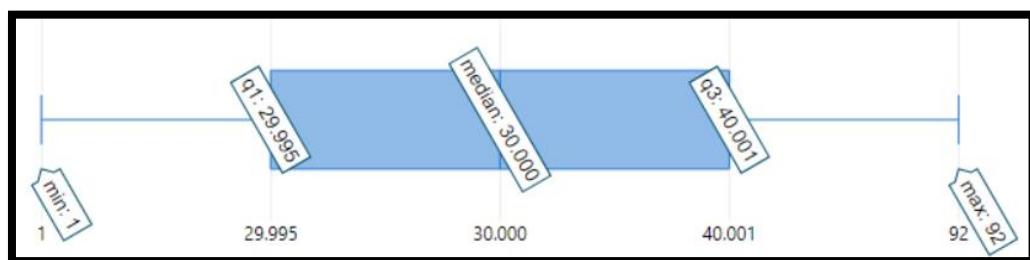
**Figura 69.** Caja de bigotes de la variable pledged.

**Fuente:** Elaboración propia.

- Duración de la campaña del proyecto (*duration*):

Los principales datos estadísticos de esta variable son los siguientes:

- Rango de valores: [1; 92]
- Media: 35.4654141132436
- Mediana: 30
- Desviación estándar: 11.84570862999998
- Varianza: 140.320812946853

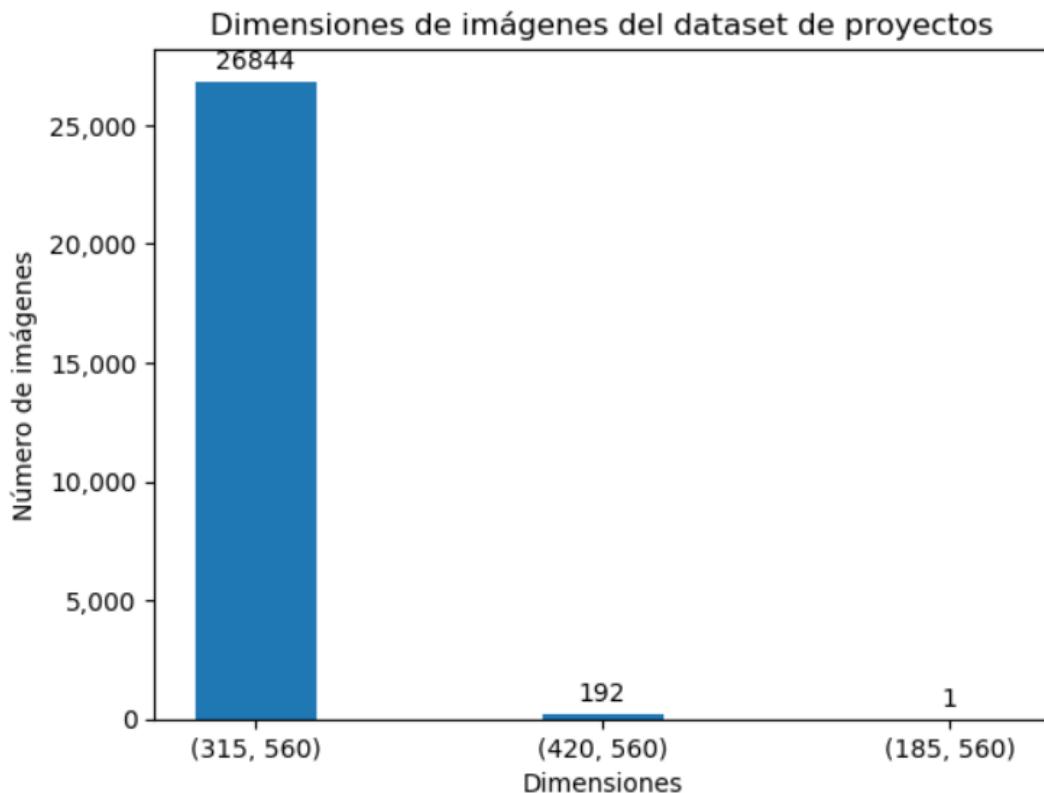


**Figura 70.** Caja de bigotes de la variable duration.

**Fuente:** Elaboración propia.

#### 4.2.2. Contenido visual

Del universo de las 27,248 imágenes de proyectos tecnológicos de Kickstarter, no se pudieron determinar las dimensiones de 211 a través de la función *shape* de la librería cv2 (OpenCV) y por ello, no se pueden re-dimensionar. Sin embargo, del resto de las 27,037 imágenes, se determinaron 3 clases de dimensiones (alto x ancho), de las cuales se distribuyen de acuerdo a la **Figura 71**.



**Figura 71.** Distribución de clases de dimensiones de imágenes de proyectos.

**Fuente:** Elaboración propia.

#### 4.2.3. Contenido textual

Del universo de 27,249 descripciones extraídas de los 27,251 proyectos en total (2 de ellos no se pudieron descargar por acceso no permitido), 638 proyectos (2% del total) no presentaron descripción alguna, es decir, estaban en blanco. Asimismo, la descripción con mayor longitud tenía 32,430 caracteres.

### 4.3. Pre-procesamiento de los conjuntos de datos

#### 4.3.1. Metainformación

Con el conjunto de datos disponible en Kaggle, se realizó el pre-procesamiento desde Colaboratory de Google Drive. A través un nuevo Token API creado desde la cuenta del usuario en la plataforma (en formato .json), se descargó el archivo .csv.

Como se mencionó anteriormente, de los 27,251 registros de proyectos en total se eliminaron 216 ya que 214 de ellos tenían imágenes que no pudieron redimensionarse y los 2 restantes no contenían descripciones, resultando al final 27,035 registros empleados.

Como se observó en la Figura N° 66, con excepción de la variable duration, las variables independientes no siguen una distribución normal y, además, presenta altas desviaciones estándar. Por ello, se normalizaron estos datos usando el escalador Min-Max (*Min-Max Scaler*) de la librería SKLearn, con el fin de reducirlos al rango entre 0 y 1. Con ello, se normalizan todos los valores de las variables al mismo rango como se aprecia en la **Figura 72**. La fórmula de este escalador es la siguiente:

$$\frac{x_i - \min(x)}{\max(x) - \min(x)}$$

**Ecuación 25.** Fórmula del escalador Min-Max.

**Fuente:** (Keen, 2017).

	count	mean	std	min	25%	50%	75%	max
<b>backers_count</b>	27035.0	0.001977	0.011176	0.0	0.000019	0.000085	0.000661	1.0
<b>goal</b>	27035.0	0.000879	0.011085	0.0	0.000050	0.000180	0.000500	1.0
<b>pledged</b>	27035.0	0.002000	0.013076	0.0	0.000001	0.000029	0.000464	1.0
<b>duration</b>	27035.0	0.378552	0.130054	0.0	0.318681	0.318681	0.428571	1.0

**Figura 72.** Estadísticas de los datos normalizados.

**Fuente:** Elaboración propia.

Una vez logrado esto, el conjunto de datos total fue separado en subconjuntos de entrenamiento (0.80) y prueba (0.20).

#### 4.3.2. Contenido visual

Con el conjunto de datos disponible en Kaggle, se realizó el pre-procesamiento desde Colaboratory de Google Drive. A través un nuevo Token API creado desde la cuenta del usuario en la plataforma (en formato .json), se descargó el archivo comprimido que contenía la carpeta de imágenes y el archivo de valores separados por coma (.csv) de los estados etiquetados de cada proyecto. A continuación, se creó la carpeta “[project\\_images](#)” en donde se almacenarían las imágenes descomprimidas.

Para poder una gran performance usando la red VGG-19, es necesario que todas las imágenes sean cuadradas (alto=ancho) y que posean las mismas dimensiones. Por ello, usando la función *resize* de la librería OpenCV, se creó una función para re-dimensionar las imágenes obteniendo como entrada sus dimensiones (alto y ancho) y retornando las mismas re-dimensionadas. Aquellas que no pudieron modificarse mediante esta función fueron descartadas del conjunto final de imágenes.

Finalmente, estas fueron redirigidas a la subcarpeta “[resized\\_images](#)” dentro de “[project\\_images](#)”, quedando un total de 27,037 imágenes con nuevas dimensiones. Se eliminaron dos imágenes cuyos id no figuran en la data de descripciones, quedando al final 27,035. Al mismo tiempo, se cargó el conjunto de datos de las etiquetas del estado de cada proyecto y se eliminaron aquellos registros cuyos id no figuraren dentro de la ruta de imágenes “[project\\_images/resized\\_images](#)”. Con esta equidad de datos entre el conjunto de imágenes y etiquetas, se procedió a generar tres subconjuntos de entrenamiento, validación y prueba para el archivo .csv de etiquetas. Las proporciones dadas fueron de 0.80, 0.10 y 0.10 respectivamente de acuerdo al octavo antecedente, basado en la predicción del éxito de proyectos crowdfunding con Deep Learning (Yu, y otros, 2018). La variable independiente del conjunto de etiquetas (*id*, la X) fue separada de la variable dependiente (*status*, la Y).

Una vez dividido el conjunto de etiquetas en estos porcentajes, se crearon tres subcarpetas (*train*, *validation* y *test*) donde las imágenes fueron redirigidas desde su ruta actual hacia la que le corresponde de acuerdo a la búsqueda de su nombre en alguno de los tres subconjuntos nuevos. Es decir, si el nombre de una determinada imagen figuraba en el conjunto de entrenamiento, su imagen sería redirigida a la ruta “[project\\_images/train](#)”.

Posterior a la organización de las imágenes en sus carpetas correspondientes, los subconjuntos de entrenamiento, validación y prueba de las imágenes serían convertidos a vectores una vez cargadas a memoria.

#### 4.3.3. Contenido textual

Al igual que el contenido visual, con el conjunto de datos disponible en Kaggle, se tenía pensado realizar al inicio el pre-procesamiento desde Colaboratory de Google Drive. Sin embargo, por el gran tamaño que representa la data total, se procedió a realizar el pre-procesamiento en Jupyter Notebook desde un servidor en Google Cloud.

En primer lugar, se cargaron las dos bases de datos que se usaron: la metainformación y las descripciones de proyectos. La primera (**Figura 73**) solo fue usada para obtener la columna de etiquetas del estado (“successful” y “failed”), la cual sería la variable dependiente Y correspondiente para el modelo, a través de la búsqueda de registros mediante columna *id* del segundo conjunto de datos. La única variable X del modelo sería la columna *description* de la segunda base de datos (**Figura 74**).

	<b>id</b>	<b>backers_count</b>	<b>name</b>	<b>blurb</b>	<b>category</b>	<b>photo</b>	<b>urls</b>	<b>city</b>	<b>cour</b>
0	1000245024	0	Safer Home	Placing furniture against traditional plugs is...	Hardware	<a href="https://ksr-ugc.imgix.net/assets/011/663/874/0...">https://ksr-ugc.imgix.net/assets/011/663/874/0...</a>	<a href="https://www.kickstarter.com/projects/homesafet...">https://www.kickstarter.com/projects/homesafet...</a>	Kamloops	
1	1000256230	0	Shreddit - Privacy on Reddit	Shreddit, a Reddit privacy tool I created and ...	Software	<a href="https://ksr-ugc.imgix.net/assets/013/466/903/0...">https://ksr-ugc.imgix.net/assets/013/466/903/0...</a>	<a href="https://www.kickstarter.com/projects/466914929...">https://www.kickstarter.com/projects/466914929...</a>	Edinburgh	
2	1000261018	3	Diposta - liberating people from their postal ...	The problem of mail: it is physical Diposta c...	Web	<a href="https://ksr-ugc.imgix.net/assets/012/071/808/1...">https://ksr-ugc.imgix.net/assets/012/071/808/1...</a>	<a href="https://www.kickstarter.com/projects/105350477...">https://www.kickstarter.com/projects/105350477...</a>	Raleigh	
3	100039820	3	Best Spray Bottle Ever - SureShot	Ever had a spray bottle that has a little bit ...	Gadgets	<a href="https://ksr-ugc.imgix.net/assets/012/009/461/3...">https://ksr-ugc.imgix.net/assets/012/009/461/3...</a>	<a href="https://www.kickstarter.com/projects/110136848...">https://www.kickstarter.com/projects/110136848...</a>	Edmonton	
4	1000468345	6	The iRNinja - Simplify your TV with a wireless...	Control your TV & Audio/Video components by pr...	Gadgets	<a href="https://ksr-ugc.imgix.net/assets/016/561/251/1...">https://ksr-ugc.imgix.net/assets/016/561/251/1...</a>	<a href="https://www.kickstarter.com/projects/5877985/i...">https://www.kickstarter.com/projects/5877985/i...</a>	Jupiter	

**Figura 73.** Conjunto de datos de metainformación de proyectos.

**Fuente:** Elaboración propia.

	<b>id</b>	<b>description</b>
0	1000245024	Purpose Safer Home ensures that electrical plu...
1	1000256230	I once had a Reddit account that was four year...
2	1000261018	Every day you go home to a mail box filled wit...
3	100039820	Funds needed for tooling. Once tooling is done...
4	1000468345	The iRNinja is a wireless 4 button keypad that...
5	1000551777	Gantish is a web-based platform for parents, w...

**Figura 74.** Conjunto de datos de descripciones de proyectos.

**Fuente:** Elaboración propia.

Se eliminaron 214 proyectos cuyas imágenes no pudieron re-dimensionarse más 2 que no presentaban descripción. Así, se formó dos conjuntos finales de X y Y de 27,035 registros cada uno (**Figura 75** y **Figura 76**).

	<b>description</b>
1	I once had a Reddit account that was four year...
2	Every day you go home to a mail box filled wit...
3	Funds needed for tooling. Once tooling is done...
4	The iRNinja is a wireless 4 button keypad that...
5	Gantish is a web-based platform for parents, w...
6	Hello, At SellEgg, our goal is to establish co...
7	<b>failed</b>

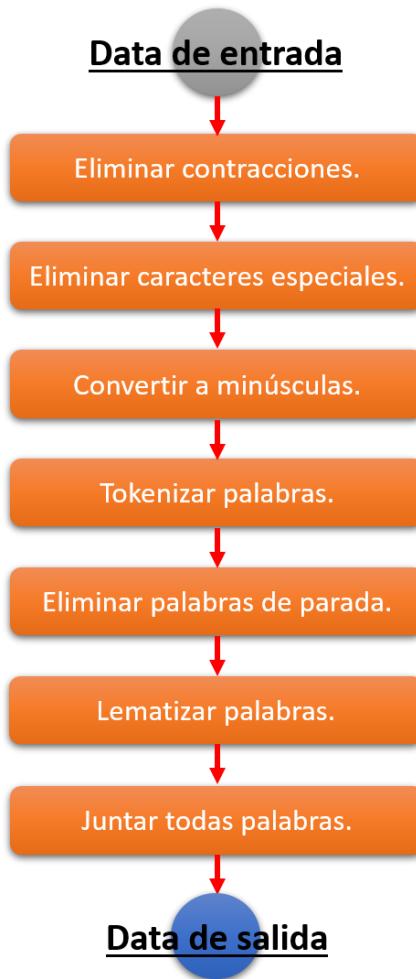
**Figura 75.** Conjunto X generado para el modelo de descripciones (izquierda).

**Fuente:** Elaboración propia.

**Figura 76.** Conjunto Y generado para el modelo de descripciones (derecha).

**Fuente:** Elaboración propia.

Para poder usar un modelo de Máquina de Vectores de Soporte (SVM) y evaluarlo con modelos de **Bolsa de Palabras** (*Bag of Words*, BoW) y **Frecuencia de Término-Frecuencia Inversa de Documento** (*Term Frequency–Inverse Document Frequency*, TF-IDF), se debe realizar una limpieza del contenido mediante el siguiente flujo de procedimientos.



**Figura 77.** Flujograma de limpieza de conjunto de datos de descripciones.

**Fuente:** Elaboración propia.

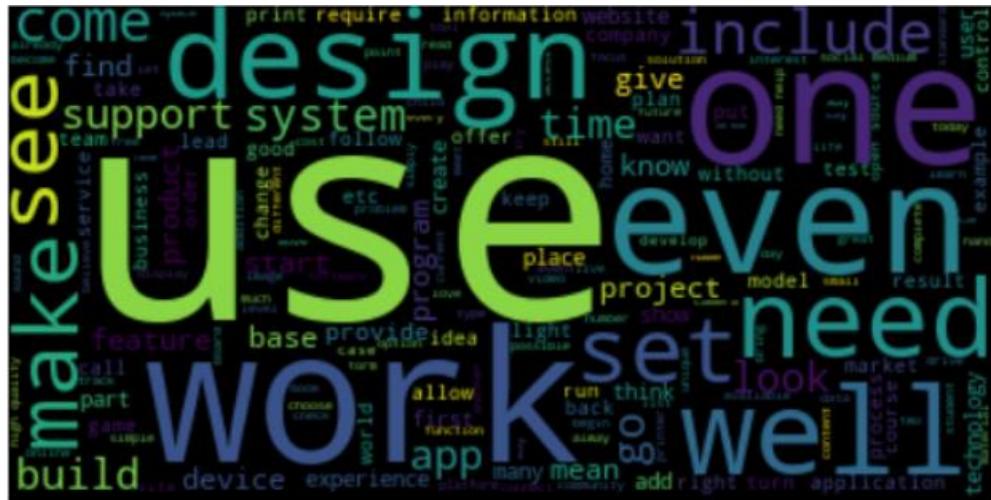
Gracias a la librería de Python NLTK (*Natural Language Toolkit*), se puede procesar conjuntos de datos basados en texto para representarlos como el lenguaje natural humano. El primer paso consiste en eliminar contracciones de palabras en inglés para obtener expresiones regulares. Luego, se procederá a eliminar caracteres especiales como signos, números, entre otros. A continuación, este nuevo conjunto se convierte a minúsculas para estandarizar todas las palabras y son separadas o *tokenizadas* para poder identificar las palabras de parada. Estas últimas son eliminadas ya que individualmente no presentan sentido. Las palabras restantes fueron lematizadas, es decir, reducidas a su forma origen en caso de haberse conjugado. Finalmente, el conjunto de palabras “limpias” fue juntado en un vector por cada proyecto, quedando como resultado la **Figura 78**.

	<b>description</b>	<b>re_contractions</b>	<b>re_special_char</b>	<b>lower_case</b>	<b>tokens</b>	<b>re_stopwords</b>	<b>lemmatization</b>	<b>text_clean</b>
1	I once had a Reddit account that was four years old with thousands of comments forever stored in...	I once had a Reddit account that was four years old with thousands of comments forever stored in...	I once had a Reddit account that was four years old with thousands of comments forever stored in...	i once had a reddit account that was four years old with thousands of comments forever stored in...	[i, once, had, a, reddit, account, that, was, four, years, old, with, thousands, of, comments, f...	[reddit, account, four, years, old, thousand, comments, forever, stored, reddit, archives, cons...	[reddit, account, four, year, old, thousand, comment, forever, store, reddit, archive, consider,...	reddit account four year old thousand comment forever store reddit archive consider see example ...
2	Every day you go home to a mail box filled with junk and even worse, if you are a traveler it is...	Every day you go home to a mail box filled with junk and even worse, if you are a traveler it is...	Every day you go home to a mail box filled with junk and even worse if you are a traveler it is ...	every day you go home to a mail box filled with junk and even worse if you are a traveler it is ...	[every, day, you, go, home, to, a, mail, box, filled, with, junk, and, even, worse, traveler, overflowing, quickly, you, are...	[every, day, go, home, mail, box, filled, junk, even, worse, traveler, overflow, quickly, mail...	[every, day, go, home, mail, box, fill, junk, even, bad, traveler, overflow, quickly, mail, hold...	every day go home mail box fill junk even bad traveler overflow quickly mail hold mail forward r...
3	Funds needed for tooling. Once tooling is done, approximately 30-45 days, bottle production can ...	Funds needed for tooling. Once tooling is done, approximately 30-45 days, bottle production can ...	Funds needed for tooling Once tooling is done approximately days bottle production can begin P...	funds needed for tooling once tooling is done approximately days bottle production can begin p...	[funds, needed, for, tooling, once, tooling, is, done, approximately, days, bottle, production, ...	[funds, needed, tooling, tooling, once, tooling, is, done, approximately, days, bottle, production, begin, patents,...	[fund, need, tool, tool, do, approximately, day, bottle, production, begin, patent, file, pendin...	fund need tool tool do approximately day bottle production begin patent fil pending huge market...
4	The iRNinja is a wireless 4 button keypad that controls your entertainment system. It has two co...	The iRNinja is a wireless 4 button keypad that controls your entertainment system. It has two co...	The iRNinja is a wireless button keypad that controls your entertainment system It has two comp...	the irninja is a wireless button keypad that controls your entertainment system it has two comp...	[the, irninja, is, a, wireless, button, keypad, that, controls, your, entertainment, system, it,...	[irninja, wireless, button, keypad, controls, entertainment, system, two, components, keypad, mo...	[irninja, wireless, button, keypad, control, entertainment, system, two, component, keypad, moun...	irninja wireless button keypad control entertainment system two component keypad mount wall base...
5	Gantish is a web-based platform for parents, where they can create beautifully designed reward c...	Gantish is a web-based platform for parents, where they can create beautifully designed reward c...	Gantish is a web based platform for parents where they can create beautifully designed reward ch...	gantish is a web based platform for parents where they can create beautifully designed reward ch...	[gantish, is, a, web, based, platform, for, parents, where, they, can, create, beautifully, designed, reward, charts, children...	[gantish, web, based, platform, parent, create, beautifully, design, reward, chart, child, simple...	[gantish, web, base, platform, parent, create, beautifully, design reward chart child simple step design...	gantish web base platform parent create beautifully design reward chart child simple step design...

**Figura 78.** Proceso de limpieza de data de descripciones.

**Fuente:** Elaboración propia.

Con el conjunto de textos limpios, se elaboró una nube de palabras (*Word Clouds*) para conocer las palabras que más frecuentan en el conjunto de datos total.



**Figura 79.** Nube de palabras del contenido textual total.

**Fuente:** Elaboración propia.

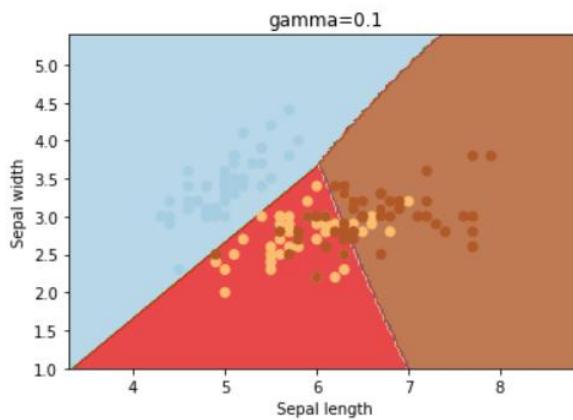
Además, se convirtió en el nuevo conjunto X y, junto al conjunto Y, se dividió cada uno en subconjuntos de entrenamiento y prueba con proporciones de 0.80/0.20 de acuerdo al octavo antecedente (Yu, y otros, 2018). Esto representó 21,628 registros de entrenamiento y 5,407 de prueba.

#### 4.4. Creación de los modelos predictivos

##### 4.4.1. Metainformación

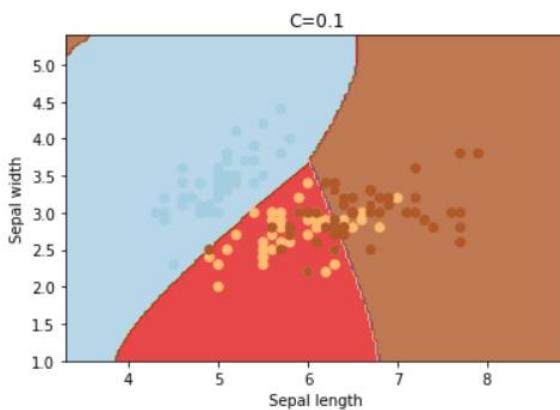
Se realizaron dos modelos para comparar los resultados de estos experimentos: Un modelo de Máquina de Vectores de Soporte (SVM) y un modelo de Red Neuronal Multicapa (MLP).

Para el primer modelo, se obtuvieron los mejores hiper-parámetros a través de una simulación del mismo con un núcleo lineal (*linear kernel*) y pesos balanceados según la clase que corresponda. Los resultados de este proceso conocido como ajuste de parámetros (*parameter tuning*) fueron que el mejor **gamma**, aquel que define qué tan recto serán los hiperplanos que contengan cada registro del conjunto de datos (Ben Fraj, 2018), fue de 0.1 y el mejor **C**, aquel que penaliza el término de error y controla la compensación entre el límite de decisión suave y la clasificación correcta de puntos de entrenamiento (Ben Fraj, 2018), fue de 0.1.



**Figura 80.** Ejemplo de parámetro  $\gamma = 0.1$

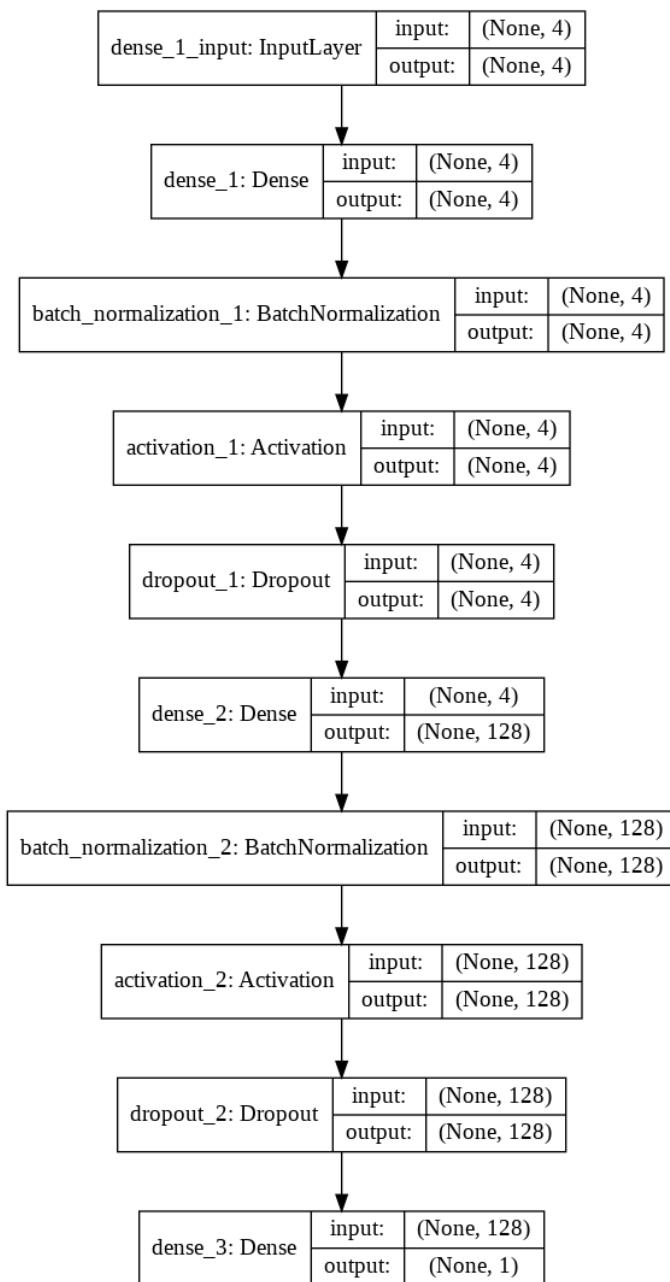
**Fuente:** (Ben Fraj, 2018)



**Figura 81.** Ejemplo de parámetro  $C = 0.1$

**Fuente:** (Ben Fraj, 2018).

Por otro lado, el segundo modelo se construyó bajo la siguiente arquitectura:



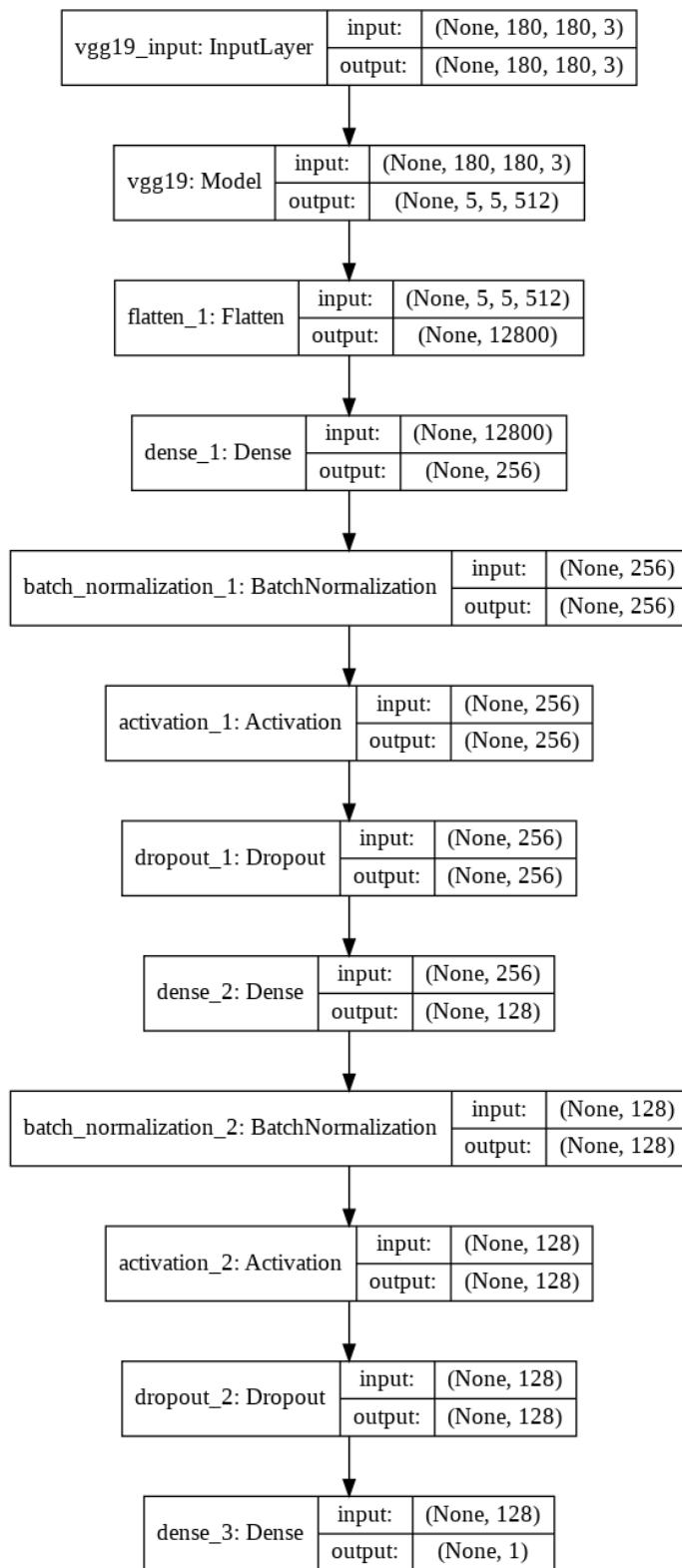
**Figura 82.** Arquitectura del modelo de Red Multicapa para la metadata.

**Fuente:** Elaboración propia.

Ya que las etiquetas de la variable dependiente *status* presenta una proporción disparaja, se usaron pesos balanceados de acuerdo a esta condición para lograr una mejor performance. Así, si el estado era “failed” (casi el 70% del total de proyectos), su peso asignado fue **0.69912077**, mientras que, si era “successful”, su peso fue **1.75551948**.

#### 4.4.2. Contenido visual

La arquitectura del modelo final usado fue la siguiente:

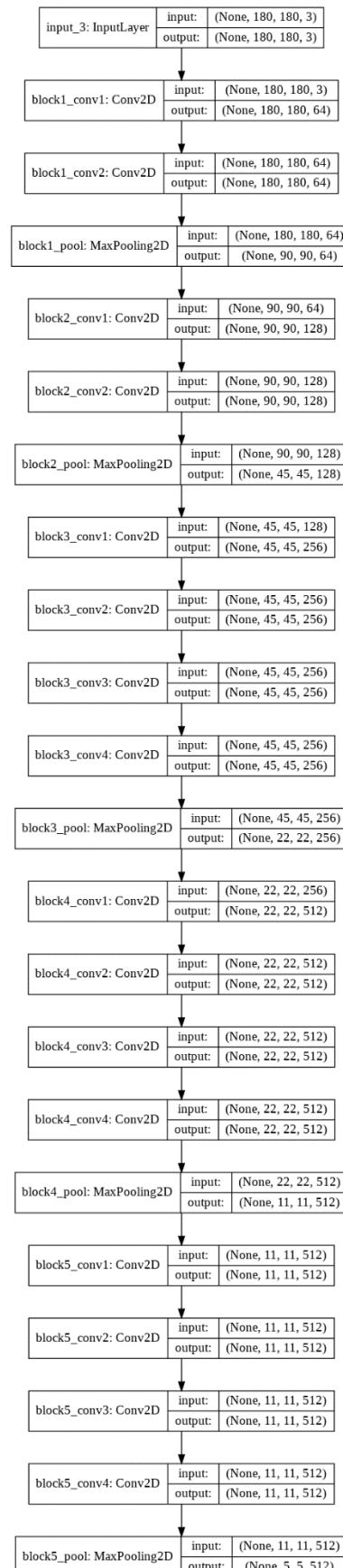


**Figura 83.** Arquitectura del modelo final para las imágenes.

**Fuente:** Elaboración propia.

Este modelo llamado **modelo de inicio** (*inception model*) se basa en el **aprendizaje por transferencia** (*Transfer learning* en inglés) ya que se está “transfiriendo” los mejores pesos y características de un modelo anterior a otro diferente con el fin de mejorar su performance. Como se aprecia en la **Figura 83**, la capa de entrada recibirá a las imágenes re-dimensionadas en 180x180 para luego reducirlas en la siguiente capa hasta 5x5. Esta última es el modelo de red neuronal convolucional VGG-19, el cual se basa en 19 capas donde hay 5 bloques: los tres primeros tienen dos capas de convolución y una de reducción cada uno, mientras que los dos bloques restantes presentan cuatro capas de convolución y una de reducción cada uno. La arquitectura se completa con cuatro capas completamente conectadas y la función de activación sigmoide, resultando en la **Figura 84**.

Para el presente trabajo, se cargaron los pesos de un modelo pre-entrenado (“*imagenet*”) basado en clasificación de imágenes, los cuales fueron añadidos al modelo VGG-19 para mejorar su performance. Luego, en las siguientes capas se empiezan a convolucionar y reducir las imágenes que reciben de entrada y al final de cada bloque se genera una imagen más reducida con sus características extraídas para servir como entrada en las capas posteriores. Finalmente, en la última capa se reducen las imágenes hasta dimensiones de 5x5 y 512 de profundidad (este último valor resulta del valor cuadrático creciente que se genera gracias a los filtros aplicados), generando en total 20,024,384 parámetros, todos entrenables.



**Figura 84.** Arquitectura del modelo VGG-19 con pesos pre-entrenados.

**Fuente:** Elaboración propia.

A continuación, regresando al modelo de inicio, todas las nuevas imágenes reducidas son aplanadas a un vector monodimensional de 12,800 características (el resultado del producto de las tres dimensiones 5x5x12). Luego, estas son reducidas nuevamente a 256 en una capa densa, donde se normalizan, se usa la función de activación y se desconectan algunos nodos para evitar redundancia y, por consiguiente, lograr mejor rendimiento. Este mismo proceso se repite una vez más, se obtiene al final una capa densa de 128 características extraídas que genera una nueva capa densa de una característica.

El total de parámetros que este modelo genera es de 23,335,617, de los cuales 3,310,465 son entrenables.

En este modelo de inicio se balancearon los pesos de acuerdo a las proporciones de los dos estados, “failed” y “successful”, los cuales fueron de **0.69808276** y **1.76209875** respectivamente. Para evitar el sobreajuste al entrenar el modelo, se usó la **parada temprana** (*Early Stopping*) que ofrece la librería de Keras. Como parámetros, se asignaron como condiciones que esta funcione en el puntaje AUC para el subconjunto de validación de imágenes, con un delta mínimo de 0.1, una paciencia de 4 eventos y modo automático.

Antes de entrenar el modelo, asimismo, se definió que la métrica principal que evaluaría la performance del mismo sería la del puntaje AUC. La función de pérdida sería “**entropía cruzada binaria**” (*binary\_crossentropy*) ya que se trata de una clasificación binaria por tener la variable *status* solo dos valores posibles. El optimizador elegido fue el **RMSprop**, especializado en dividir el gradiente entre un promedio de su magnitud reciente, con una ratio de aprendizaje de 0.000001 y una rho de 0.7.

#### 4.4.3. Contenido textual

Como se comentó anteriormente, se usaron las medidas *Bolsa de Palabras (BoW)* y *TF-IDF* para determinar la frecuencia de una serie de palabras de un vocabulario dentro de un texto.

Antes de asignar los conjuntos de entrenamiento y prueba para BoW, se determinó un vocabulario de un total de 183,249 palabras únicas.

```
{'valerij': 166897, 'carat': 25637, 'epiony': 50944, 'cloudscript': 29846, 'awadhi': 14768, 'normalement': 107359, 'sinew': 141573, 'archivio': 10569, 'psoriatic': 125211, 'newport': 106147, 'fluorosalan': 58161, 'solidary': 144121, 'preemptive': 121956, 'nomination': 107130, 'praktischen': 121715, 'vpr': 170787, 'mender': 97538, 'теплами': 181131, 'clubben': 29874, 'clemente': 29523, 'reelle': 129522, 'qu': 126372, 'techtimes': 153244, 'actionscript': 1795, 'boardbeen': 20822, 'dunedin': 45274, 'base': 16405, 'qfoods': 126217, 'hcmiu': 69557, 'shiro': 140324, 'orgdoesn': 112067, 'logically': 91744, 'activityif': 1877, 'mutable': 103255, 'distasteful': 42880, 'lesscontrol': 89436, 'cosmakerspace': 34980, 'diapason': 41387, 'theusbconnectoris microcousbypeandonlysupportchargefunction': 156892, 'bungale': 23441, 'testdates': 154233, 'prolems': 124319, 'terrific': 154154, 'robotwillmake': 133417, 'diskussion': 42613, 'munchie': 102950, 'stimulus': 148042, 'futurelearn': 61431, 'handyhuelle n': 68624, 'toshiiro': 159773, 'rené': 130838, 'regrese': 129923, 'начало': 180687, 'plugfone': 119665, 'fatherrecognised': 55501, 'imitation': 75096, 'organisatorischen': 111996, 'downloadson': 44232, 'mortage': 101675, 'redbot': 129309, 'studioand': 148944, 'tactfully': 151874, 'atimpacthub': 12765, 'recibirlodescarga': 128867, 'knapitsch': 85534, 'définissez': 45900, 'koblen': 85710, 'concordance': 32600, 'wpanel': 175861, 'sikorsky': 141146, '正面': 182728, 'colorants': 30799, 'invidual': 79612, 'payeurs': 115749, 'enamel': 49318, 'sonnenbedingte': 144528, 'sqf': 146566, 'générales': 67784, 'therregular': 156398, 'ultérieur': 163563, 'icebreaker': 73951, 'instash': 78150, 'prototypedemonstration': 124629, 'auswirkungen': 13921, 'andhow': 7188, 'carabineer': 25603, 'zalite': 178093, 'thatguypproductions': 154657, 'ln': 91469, 'unintended': 164605, 'orthopedist': 112387, 'cosideration': 34971, 'preis': 122044, 'threelockstarterspreviously': 157530, 'barium': 16034, 'isconvenience': 80307, 'piñones': 118907, 'recuse': 129264, 'listeninc': 90950, 'erguation': 51300, 'prevén': 122577, 'lifehacker': 90071, 'generically': 63256, 'active': 109022, 'gesetz': 63859, 'dessinés': 40608, 'sourcevme': 145032, 'verilog': 168108, 'besoin': 18494, 'skulptsynthesiser': 142267, 'theirmarrige': 155642, 'fashnerd': 55417, 'winterstein': 174553, 'bluerock': 20672, 'pe': 116497, 'yourmonav': 177602, 'resóns': 121253, 'postponement': 121146, 'cambiamos': 24706, 'fashiond': 55402, 'johwalle'}
```

**Figura 85.** Muestra del vocabulario de palabras de las descripciones.

**Fuente:** Elaboración propia.

Este vocabulario se obtuvo al vectorizar y contar las palabras del conjunto de textos limpio con el mencionado modelo.

Asimismo, se repitió el mismo proceso para construir la métrica TF-IDF. Si bien ambos calculan la frecuencia de palabras, estos se diferencian en cómo lo hacen. El segundo se basa en la comparación del número de veces que una palabra aparece en un documento con el número de documentos en los que la palabra aparece.

Para el caso de BoW, este algoritmo cuenta si un término aparece en un documento. Si ocurre esto, asigna un contador que irá aumentando conforme se encuentren más del mismo término. Este contador se mantendrá en 0 si es que el término no logra aparecer y el término será buscado en el siguiente documento, como en el ejemplo de la **Figura 86**.

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1			1	1	1				
Doc 3						1	1	1	2	1

**Figura 86.** Ejemplo de funcionamiento del método BoW.

**Fuente:** (Calderon, 2017).

Mientras que por el lado de TF-IDF, esta se calcula mediante la fórmula:

$$TF - IDF = TF(t, d) * IDF(t)$$

**Ecuación 26.** Fórmula del TF-IDF.

**Fuente:** (Can Tayiz, 2019).

Donde  $t$  representa el número de veces del término,  $d$  es el número de apariciones en un documento y la función  $IDF(t)$  es la frecuencia inversa de un documento determinada por:  $\log \frac{1 + \text{número de documentos}}{1 + \text{freq.del documento del término } t} + 1$

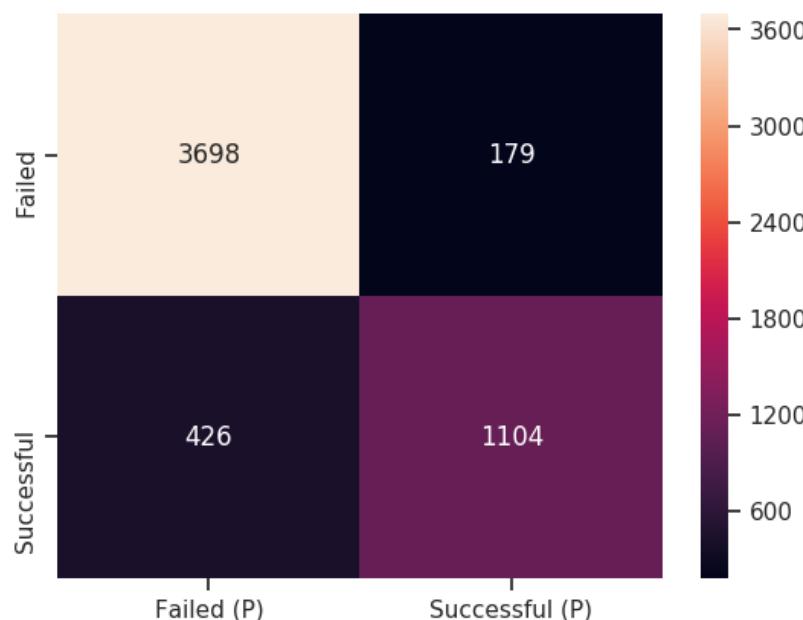
Después se creó un modelo de Máquina de Vectores de Soporte (SVM) para entrenar las dos métricas de frecuencia de palabras usando un núcleo lineal (linear kernel) y un C de 10.

## 5. CAPÍTULO V: ANÁLISIS Y DISCUSIÓN DE RESULTADOS

### 5.1. Metainformación

Tanto el primer como el segundo modelo fueron evaluados por las mismas 4 métricas del décimo antecedente: precisión (*precision*), sensibilidad (*recall*), puntaje F1 (*f1-score*) y puntaje AUC (*AUC score*). Estos valores son calculados a partir de la matriz de confusión de cada uno.

La matriz de confusión del primer modelo (SVM) fue la siguiente:



**Figura 87.** Matriz de confusión del modelo SVM para la metadata.

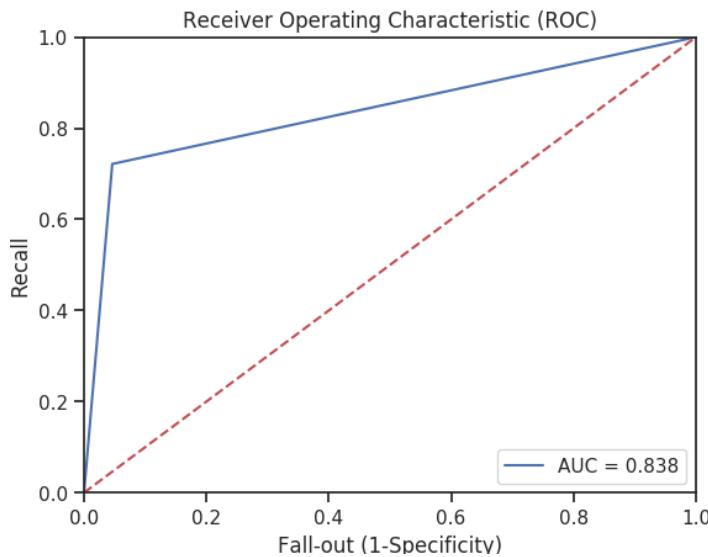
**Fuente:** Elaboración propia.

La anterior matriz muestra los resultados de valores reales y predichos por el modelo usando el subconjunto de prueba (5,407 registros: 3,877 proyectos fracasados y 1,530 proyectos exitosos). De la misma, se deduce lo siguiente:

- Los verdaderos negativos (proyectos fracasados) presentan un nivel muy alto de clasificación por parte del modelo (3,698 proyectos fracasados predichos correctamente de 3,877 proyectos fracasados reales), es decir, se tiene altas posibilidades casi perfectas de clasificar bien cuando un proyecto fracasará en ser financiado a partir de su metainformación; asimismo, los verdaderos positivos (proyectos exitosos) presentan también una alta performance (1,104 proyectos exitosos predichos correctamente de 1,530 proyectos exitosos reales).
- Los niveles de falsos negativos (179 proyectos fracasados clasificados incorrectamente de 3,877 proyectos fracasados reales) y falsos positivos (426

proyectos exitosos clasificados incorrectamente de 1,530 proyectos exitosos reales) son muy bajos, es decir, el modelo tiene pocas posibilidades de clasificar erróneamente el estado final de un proyecto a partir de su metainformación.

El gráfico del puntaje AUC bajo la curva ROC se representó como:

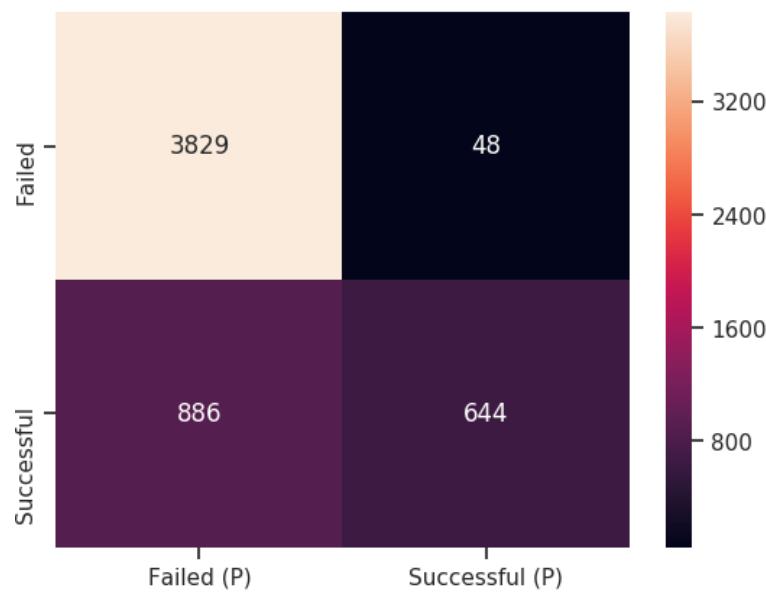


**Figura 88.** Puntaje AUC del modelo SVM para la metadata.

**Fuente:** Elaboración propia.

Este resultado de  $AUC = 0.838$  indica que el poder discriminante del modelo SVM entre las dos clases es excelente, ya que se encuentra dentro del rango [0.8; 0.9] (Britos, García Martínez, Hossian, & Sierra, 2006).

Por otro lado, la matriz de confusión del segundo modelo (Red Neuronal Multicapa) fue la siguiente:



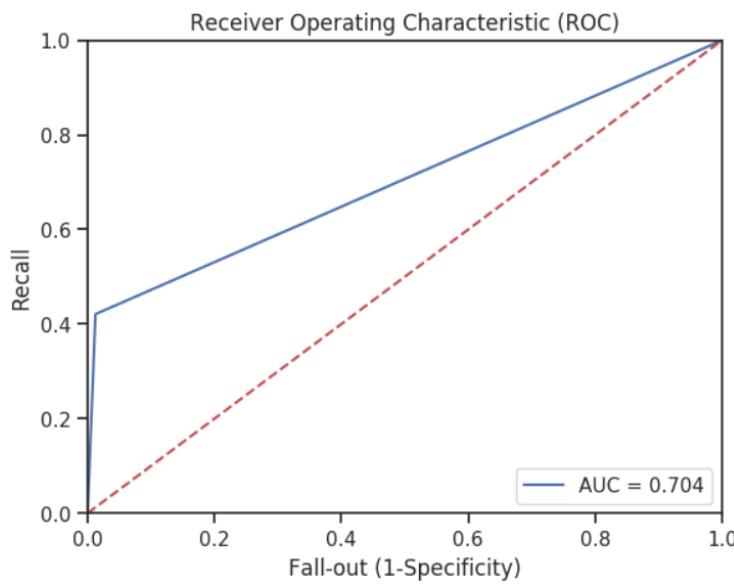
**Figura 89.** Matriz de confusión del modelo MLP para la metadata.

**Fuente:** Elaboración propia.

Conservando los mismos valores de proyectos fracasados (3,877) y exitosos (1,530) en el subconjunto de prueba, se deduce lo siguiente de la matriz:

- Los verdaderos negativos (proyectos fracasados) presentan un nivel casi perfecto de clasificación por parte del modelo (3,829 proyectos fracasados predichos correctamente de 3,877 proyectos fracasados reales), es decir, es casi seguro que el modelo siempre clasificará correctamente cuando un proyecto fracasará a partir de su metainformación; en contraste con los verdaderos positivos (proyectos exitosos) que, comparándolo con el modelo de SVM, su predicción correcta de proyectos exitosos se reduce a menos de la mitad (644 proyectos exitosos predichos correctamente de 1,530 proyectos exitosos reales).
- Los niveles de falsos negativos (48 proyectos fracasados clasificados incorrectamente de 3,877 proyectos fracasados reales) son casi nulos, es decir, es poco probable que el modelo clasifique erróneamente cuando un proyecto fracasará. Sin embargo, los falsos positivos (886 proyectos exitosos clasificados incorrectamente de 1,530 proyectos exitosos reales) superan la mitad de su total, lo cual indica que, en más de la mitad de ocasiones, el modelo se equivocará en predecir correctamente cuando un proyecto tendrá éxito a partir de su metainformación.

El gráfico del puntaje AUC bajo la curva ROC se representó como:

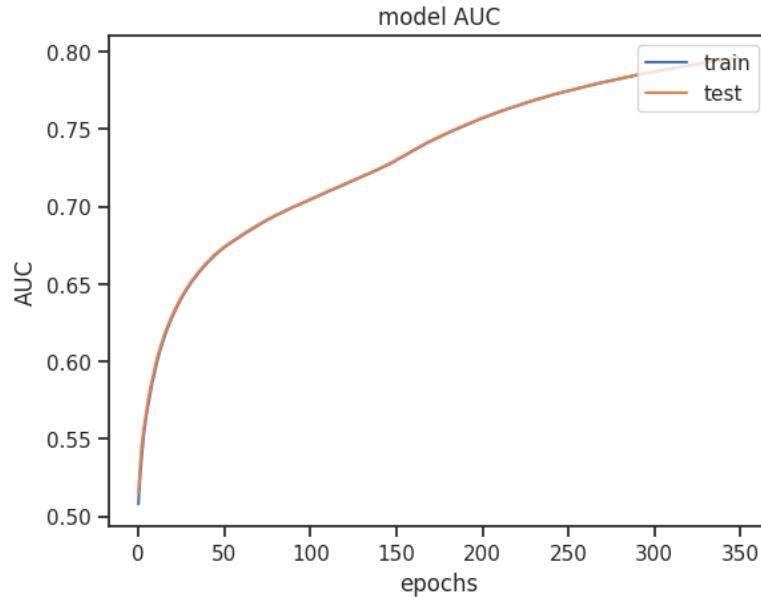


**Figura 90.** Puntaje AUC del modelo MLP para la metadata.

**Fuente:** Elaboración propia.

Este resultado de  $AUC = 0.704$  indica que el poder discriminante del modelo SVM entre las dos clases es aceptable, ya que se encuentra dentro del rango [0.7; 0.8] (Britos, García Martínez, Hossian, & Sierra, 2006).

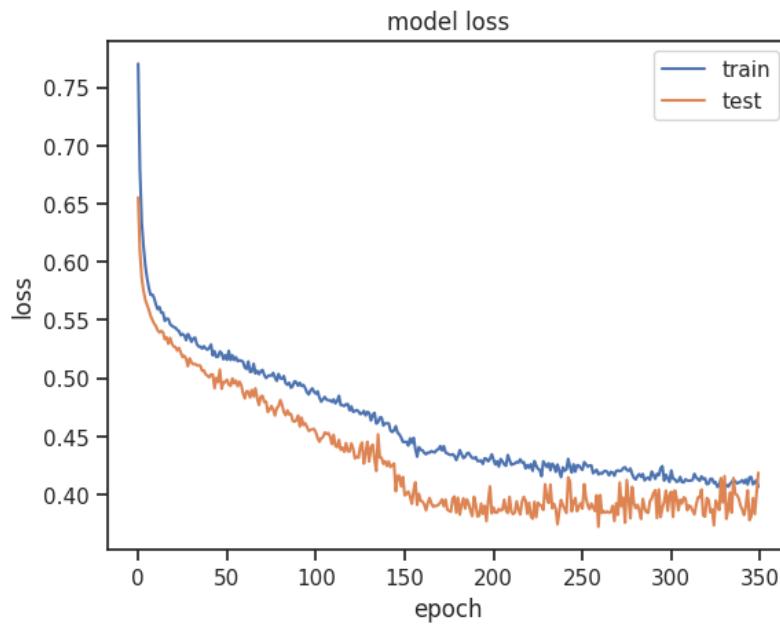
Si se le da 350 épocas para entrenar el modelo, el puntaje del AUC aumenta considerablemente tanto para el subconjunto de entrenamiento como para el de validación, convirtiéndolo en un excelente modelo.



**Figura 91.** Puntaje AUC del modelo con 350 épocas.

**Fuente:** Elaboración propia.

Si bien este modelo no presenta sobreajuste por tener pérdidas muy parejas tanto en el subconjunto de entrenamiento como en el de prueba, como se observa en la **Figura 92**, sus niveles bordean casi el 40%. Aun así, estos valores resultan ser aceptables.

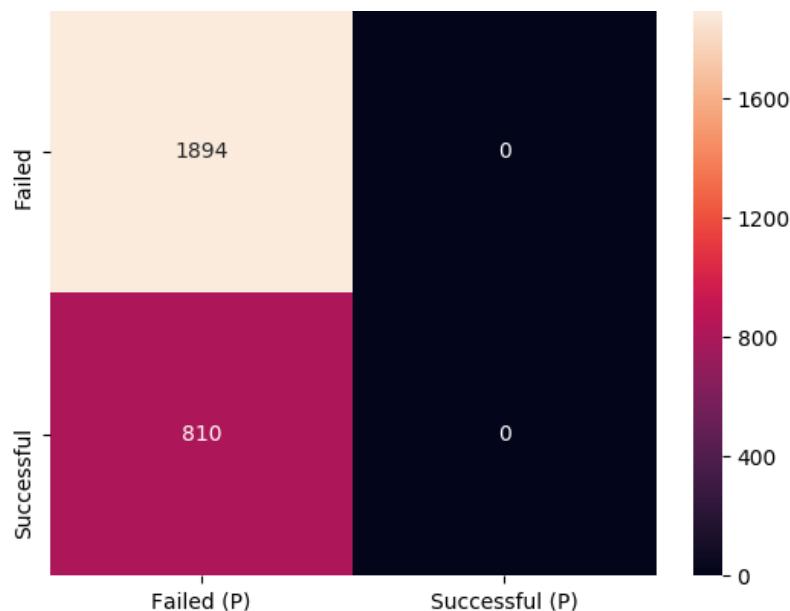


**Figura 92.** Pérdida del modelo MLP con 350 épocas.

**Fuente:** Elaboración propia.

## 5.2. Contenido visual

La matriz de confusión del modelo de inicio que contiene el VGG-19 fue:



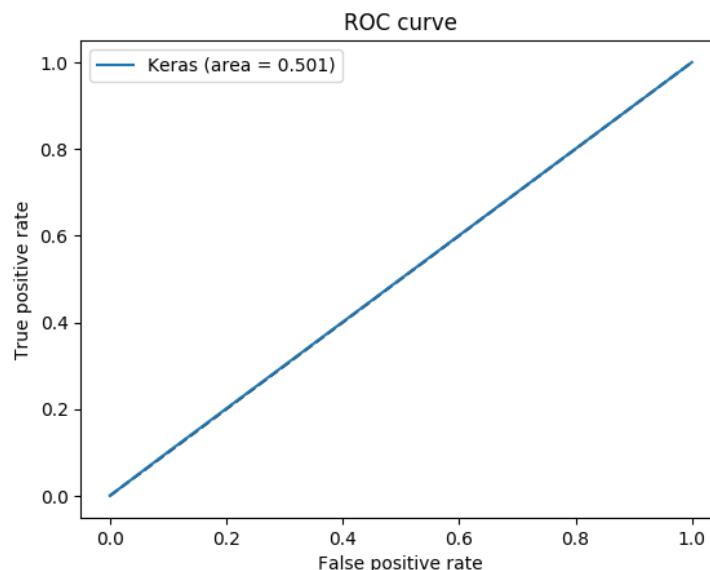
**Figura 93.** Matriz de confusión del modelo VGG-19 para las imágenes.

**Fuente:** Elaboración propia.

La anterior matriz muestra los resultados de valores reales y predichos por el modelo usando el subconjunto de prueba (2,704 registros: 1,894 proyectos fracasados y 810 proyectos exitosos). De la matriz anterior, se deduce lo siguiente:

- Los verdaderos negativos (proyectos fracasados) presentan un nivel perfecto de clasificación por parte del modelo (1,894 proyectos fracasados predichos correctamente de 1,894 proyectos fracasados reales), es decir, este siempre acertará al clasificar un proyecto fracasado a partir de su imagen principal. Sin embargo, existe un gran problema con los verdaderos positivos (proyectos exitosos) ya que el modelo no es capaz de predecir alguna vez cuando un proyecto será exitoso a partir de su imagen (0 proyectos exitosos predichos correctamente de 810 proyectos exitosos reales).
- Si bien el modelo no presenta falsos negativos (0 proyectos fracasados clasificados incorrectamente de 1,894 proyectos fracasados reales) lo cual significa que nunca clasificará erróneamente un proyecto fracasado, el nivel de falsos positivos es el máximo (810 proyectos exitosos clasificados incorrectamente de 810 proyectos exitosos reales), indicando que el modelo siempre se equivocará clasificando proyectos exitosos.

Evaluando el modelo con el puntaje AUC (área bajo la curva ROC), este no supera el 50% como se observa a continuación.

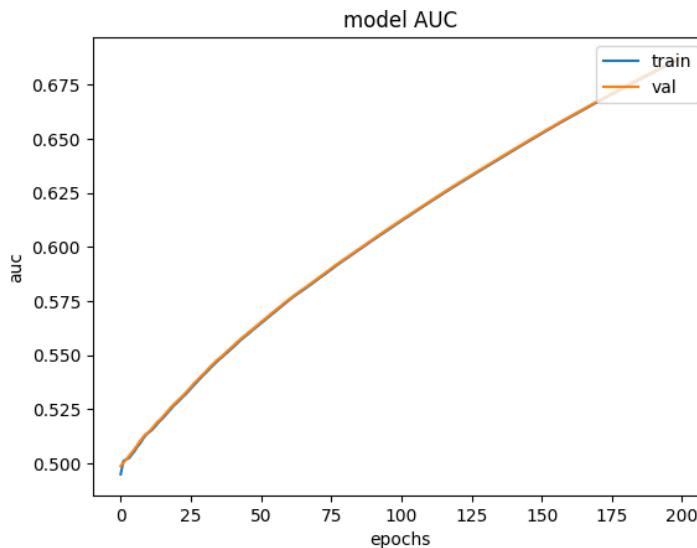


**Figura 94.** Puntaje del área AUC bajo la curva ROC del modelo de imágenes.

**Fuente:** Elaboración propia.

Esto indica que el modelo no es capaz de clasificar correctamente entre un proyecto exitoso y uno fracasado a partir de su imagen principal.

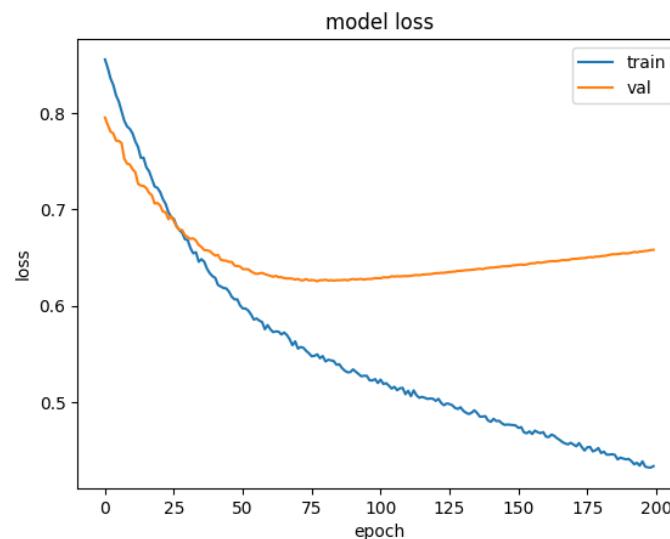
Se realizaron dos pruebas independientes usando dos métricas distintas. Los resultados para la primera prueba (usando el puntaje AUC) fueron los siguientes:



**Figura 95.** Puntaje AUC del modelo con 250 épocas para las imágenes.

**Fuente:** Elaboración propia.

Como se observa en la anterior figura, entrenando con 250 épocas al modelo se logra obtener un valor aceptable de puntaje AUC tanto para el subconjunto de entrenamiento como para el de validación. Sin embargo, con los niveles de pérdida ocurre lo siguiente:

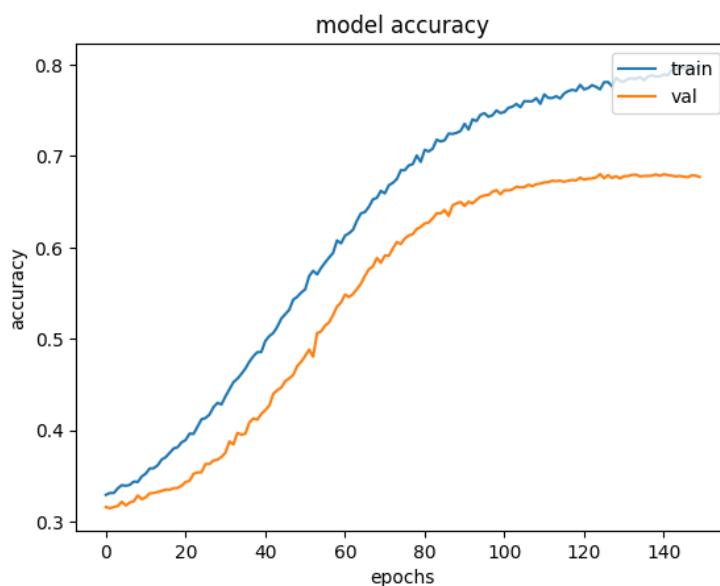


**Figura 96.** Puntajes de pérdida del AUC con 250 épocas para las imágenes.

**Fuente:** Elaboración propia.

En la anterior figura se observa un sobreajuste (*overfitting*), es decir, mayor pérdida del subconjunto de validación que el de entrenamiento, aproximadamente a partir de las 50 épocas de entrenamiento del modelo. Esto normalmente ocurre cuando un modelo solo se ajusta a aprender casos particulares que se le enseña y es incapaz de reconocer nuevos datos de entrada (Bagnato, 2017) debido a que todas las imágenes son totalmente diferentes y sin ninguna relación entre ellas.

En la segunda prueba, se obtuvieron los siguientes resultados usando la exactitud como métrica de evaluación del modelo.

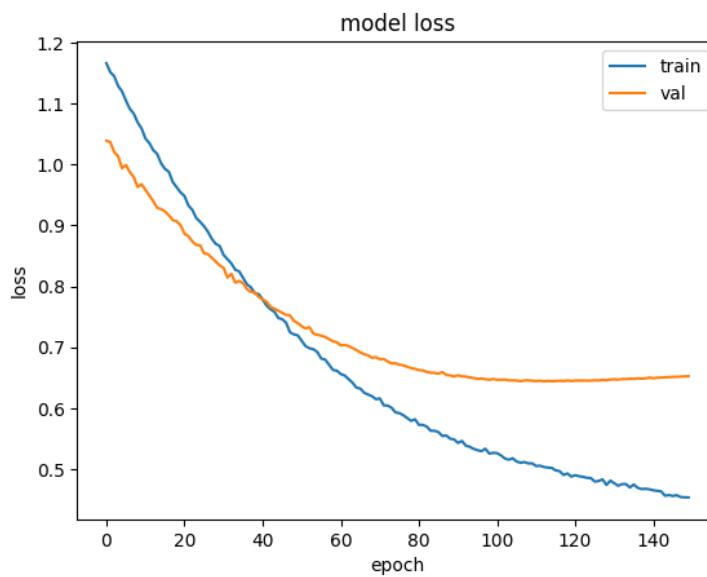


**Figura 97.** Puntaje de exactitud del modelo con 150 épocas para las imágenes.

**Fuente:** Elaboración propia.

Con 100 épocas menos que la primera prueba, el puntaje de exactitud para el modelo bordea el 80% para el subconjunto de entrenamiento y casi el 70% para el de validación.

Sin embargo, después de las 40 épocas aparece un sobreajuste en su nivel de pérdidas como se distingue a continuación.

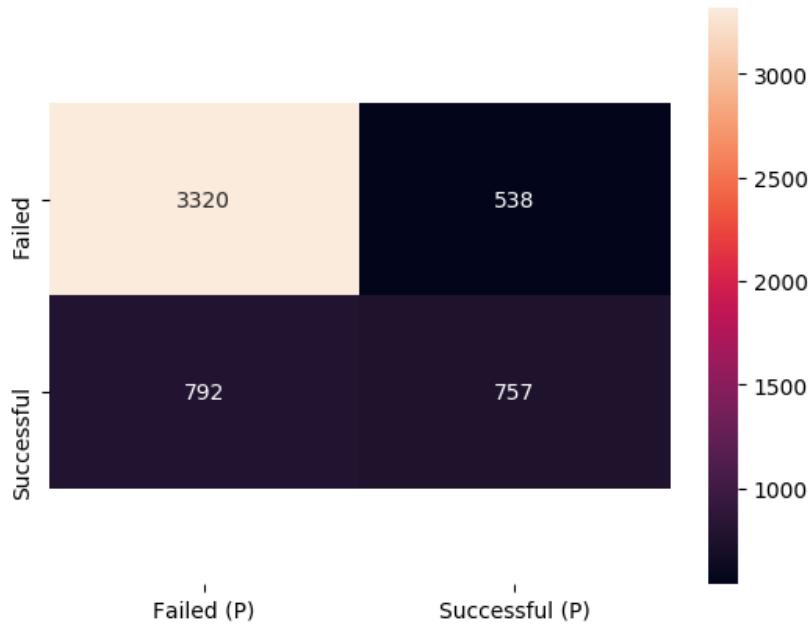


**Figura 98.** Puntaje de pérdida de exactitud con 150 épocas para las imágenes.

**Fuente:** Elaboración propia.

### 5.3. Contenido textual

La matriz de confusión del modelo de SVM usando TF-IDF fue:



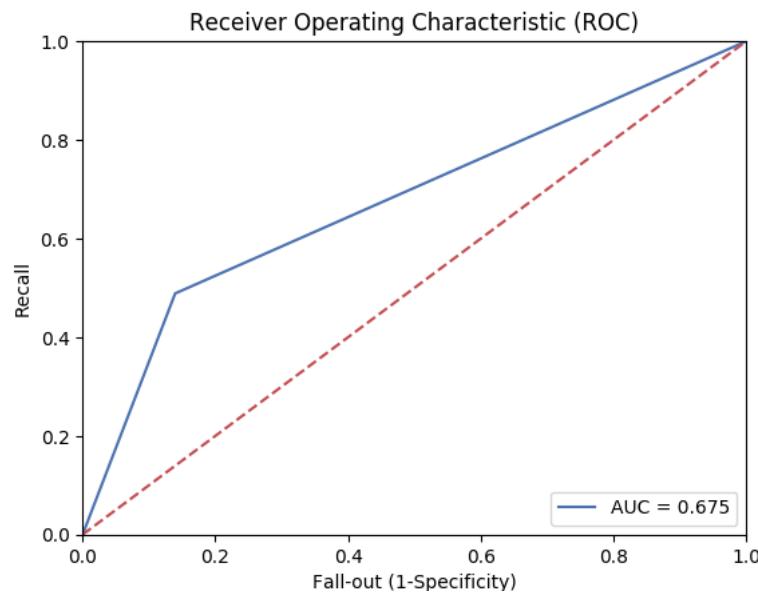
**Figura 99.** Matriz de confusión del modelo de SVM usando TF-IDF para las descripciones.

**Fuente:** Elaboración propia.

La anterior matriz muestra los resultados de valores reales y predichos por el modelo usando el subconjunto de prueba (5,407 registros: 3,858 proyectos fracasados y 1,549 proyectos exitosos). De la misma, se deduce lo siguiente:

- Los verdaderos negativos (proyectos fracasados) presentan un nivel alto de clasificación por parte del modelo (3,320 proyectos fracasados predichos correctamente de 3,858 proyectos fracasados reales), es decir, se tiene altas probabilidades casi perfectas de clasificar bien cuando un proyecto fracasará en ser financiado a partir de su descripción; en contraste con los verdaderos positivos (proyectos exitosos) ya que apenas casi la mitad de registros (757 proyectos exitosos predichos correctamente de 1,549 proyectos exitosos reales) fueron clasificados correctamente, a pesar que los pesos de sus clases fueron balanceados para lograr equidad en la fase de entrenamiento.
- Los niveles de falsos negativos (538 proyectos fracasados predichos incorrectamente de 3,858 proyectos fracasados reales) y falsos positivos (792 proyectos exitosos predichos incorrectamente de 1,549 proyectos exitosos reales) son bajos, es decir, el modelo tiene pocas probabilidades de clasificar mal un proyecto como exitoso o fracasado a partir de su descripción.

El gráfico del puntaje AUC bajo la curva ROC se representó como:

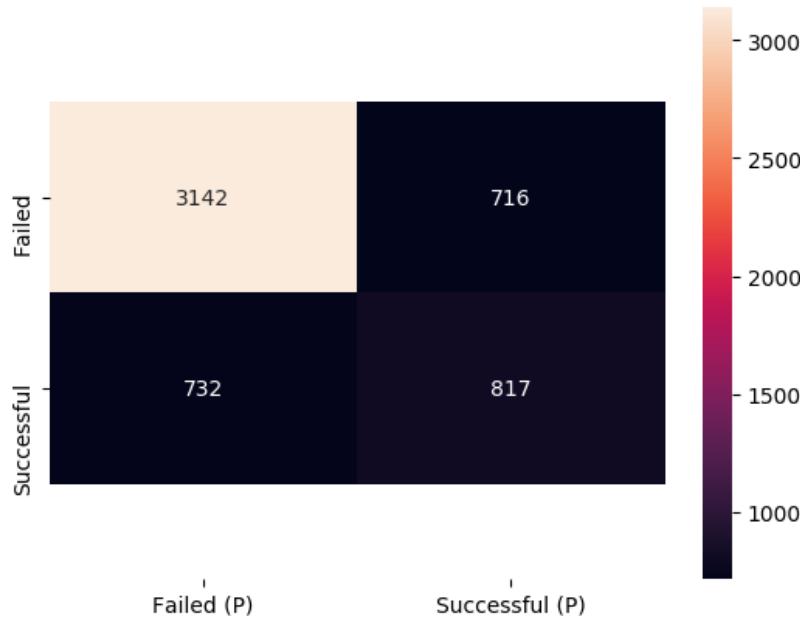


**Figura 100.** Puntaje AUC del modelo SVM usando TF-IDF para las descripciones.

**Fuente:** Elaboración propia.

Este resultado de  $AUC = 0.675$  indica que el poder discriminante del modelo SVM usando TF-IDF entre las clases es bueno mas no aceptable, ya que se encuentra dentro del rango [0.6; 0.7] (Britos, García Martínez, Hossian, & Sierra, 2006).

Por otro lado, la matriz de confusión del modelo de SVM usando BoW fue:



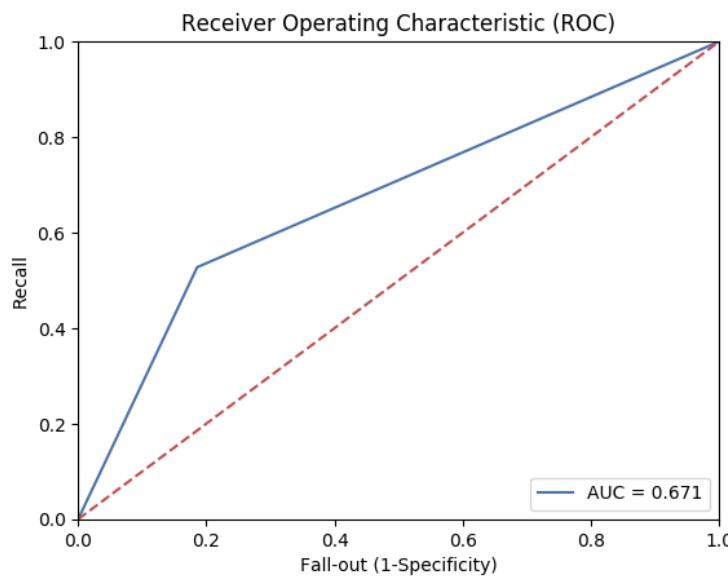
**Figura 101.** Matriz de confusión del modelo de SVM usando BoW para las descripciones.

**Fuente:** Elaboración propia.

La anterior matriz muestra los resultados de valores reales y predichos por el modelo usando el subconjunto de prueba (5,407 registros: 3,858 proyectos fracasados y 1,549 proyectos exitosos). De la misma, se deduce lo siguiente:

- Al igual que el modelo SVM usando TF-IDF, este modelo clasifica muy bien los verdaderos negativos (3,142 proyectos fracasados clasificados correctamente de 3,858 proyectos fracasados reales) y medianamente los verdaderos positivos (817 proyectos exitosos clasificados correctamente de 1,549 proyectos exitosos reales).
- Los niveles de falsos negativos (716 proyectos fracasados clasificados incorrectamente de 3,858 proyectos fracasados reales) y falsos positivos (732 proyectos exitosos clasificados incorrectamente de 1,549 proyectos exitosos reales) en promedio de este modelo de descripciones SVM son similares a las de su par que usa TF-IDF.

El gráfico del puntaje AUC bajo la curva ROC se representó como:



**Figura 102.** Puntaje AUC del modelo SVM usando BoW para las descripciones.

**Fuente:** Elaboración propia.

Este resultado de  $AUC = 0.671$  indica que el poder discriminante del modelo SVM usando BoW entre las dos clases es bueno mas no aceptable, ya que se encuentra dentro del rango [0.6; 0.7] (Britos, García Martínez, Hossian, & Sierra, 2006). Tanto este como el otro modelo basado en la descripción del proyecto presentan una probabilidad cercana al 68% de distinguir a nivel general entre un proyecto fracasado y uno exitoso.

Finalmente, se elaboró un cuadro comparativo de resultados según las métricas del décimo antecedente para todos los modelos construidos en la **Tabla 14**.

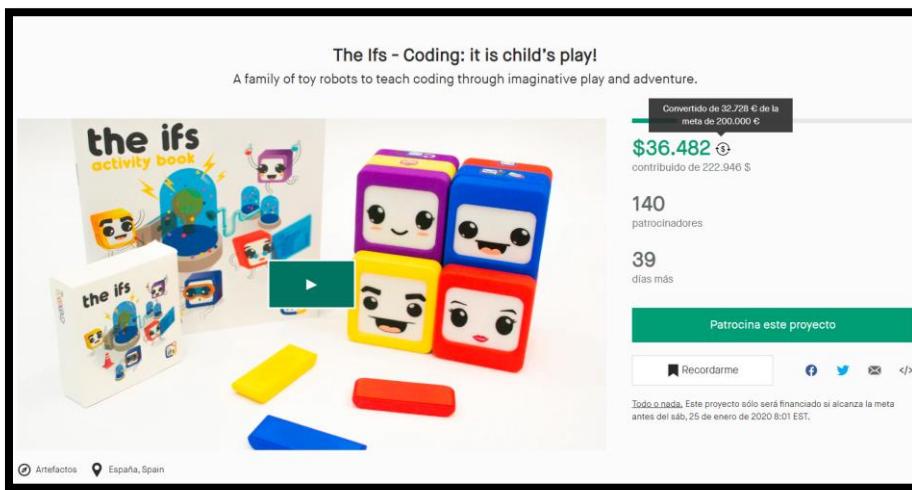
	Metainformación		Imágenes	Descripciones	
	SVM	MLP	VGG-19	SVM con TF-IDF	SVM con BoW
<b>Exactitud</b>	0.888108	0.827261	0.700444	0.754023	0.732199
<b>AUC</b>	0.837699	0.704267	0.501000	0.674626	0.670924
<b>Precisión</b>	0.860483	0.930636	0.000000	0.584556	0.532942
<b>Sensibilidad</b>	0.721569	0.420915	0.000000	0.532349	0.530175
<b>Puntaje F1</b>	0.784927	0.579658	0.000000	0.349252	0.341849
<b>Tiempo de ejecución</b>	00:00:16.22	00:39:30.15	02:43:19.26	03:42:53.30	04:25:49.16

**Tabla 14.** Comparación de los resultados de todos los modelos.

**Fuente:** Elaboración propia.

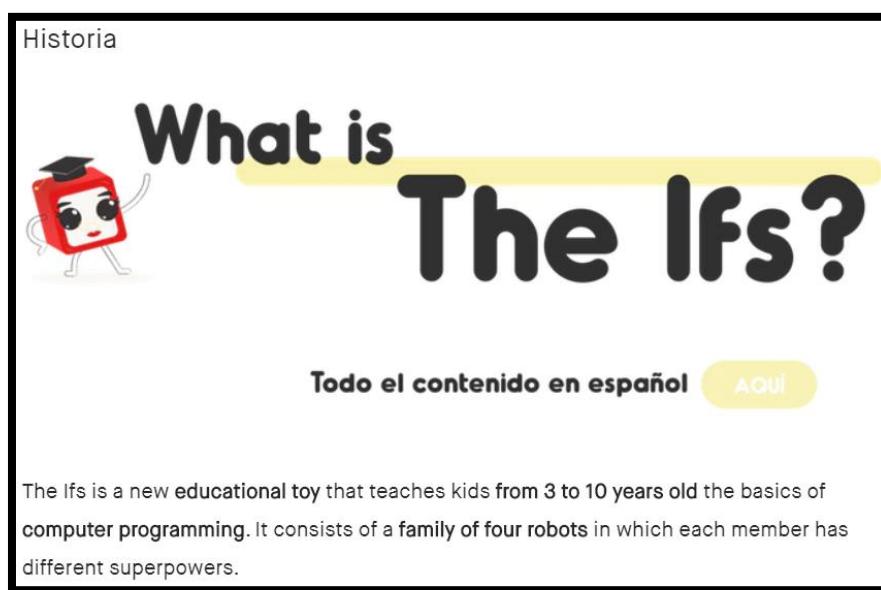
#### 5.4. Demostración de modelos

Después de guardar todos los modelos respectivos para la metainformación, descripción e imagen del proyecto, se construyó una demostración con cada uno de ellos para predecir el estado final de un proyecto aleatorio de Kickstarter, así como determinar su probabilidad de éxito de financiamiento. En las **Figura 103** y **Figura 104** se aprecian las características basadas en metainformación, imagen y descripción respectivamente.



**Figura 103.** Imagen y metainformación de proyecto de muestra a predecir su estado.

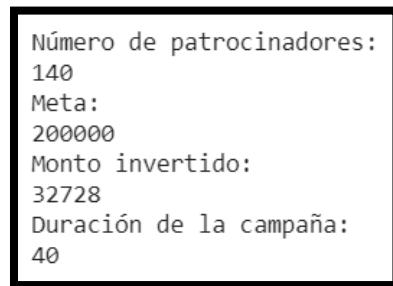
Fuente: (Latorre, 2019).



**Figura 104.** Parte de la descripción del proyecto de muestra a predecir su estado.

Fuente: (Latorre, 2019).

Desde Google Colaboratory, se cargaron los modelos guardados. Para la demo del modelo de metainformación, esta solo preguntó por las 4 variables (**Figura 105**) y a partir de ellas, emitió su predicción.



**Figura 105.** Demo del modelo de metainformación con datos del proyecto de muestra.

**Fuente:** Elaboración propia.

Para la demo del modelo de imágenes, se recibió como entrada la URL de la imagen del proyecto de muestra, después fue abierta y re-dimensionada a 180x180 (**Figura 106**) y finalmente convertida a vector de sus características para que el modelo entrenado emita su predicción.



**Figura 106.** Demo del modelo de imágenes con la imagen re-dimensionada del proyecto de muestra.

**Fuente:** Elaboración propia.

Finalmente, para la demo del modelo de descripciones, se recibió como entrada a la descripción del proyecto (**Figura 107**), posteriormente esta fue “limpiada” siguiendo el mismo proceso de pre-procesamiento de texto del flujo de trabajo de la **Figura 77**. Flujo de trabajo de limpieza de conjunto de datos de descripciones. y luego se vectorizó la lista de palabras del conjunto final de texto (se debe cargar previamente los vocabularios

entrenados tanto del modelo SVM con TF-IDF como con BoW respectivamente) para predecir el estado del proyecto.

**Escriba una descripción para la campaña de su proyecto:**  
 The Ifs is a new educational toy that teaches kids from 3 to 10 years old the basics of computer programming. It consists of a fami

**Figura 107.** Demo del modelo de descripciones con la descripción del proyecto de muestra.

**Fuente:** Elaboración propia.

Los resultados que se obtuvieron de este experimento se detallan en la **Tabla 15**.

	Metainformación		Imágenes	Descripciones		
	SVM	MLP		VGG-19	SVM con TF-IDF	SVM con BoW
Predictión de estado	Failed	Failed	Successful	Successful	Successful	Successful
Probabilidad de éxito	0.00%	0.00%	51.19%	50.99%	42.67%	

**Tabla 15.** Resultados de las demos de modelos entrenados con proyecto de muestra.

**Fuente:** Elaboración propia.

## 6. CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

### 6.1. Conclusiones

De acuerdo a los resultados mostrados en la **Tabla 14**, los modelos basados en metainformación, imágenes y descripciones tuvieron aceptables performances medidos con la exactitud. Para el caso de la metainformación, tanto la Máquina de Vectores de Soporte como el Perceptrón Multicapa obtuvieron niveles de exactitud mayor a 80% (0.8881 y 0.8273 respectivamente), lo cual significa que ambos son excelentes modelos clasificadores de proyectos basándose en esta característica de un proyecto. Por el lado de las imágenes, su nivel de exactitud alcanzó un poco más de 70% (0.7004), mientras que los dos modelos de descripciones bordean el 75% (0.7540 para SVM con TF-IDF y 0.7322 para SVM con BoW), indicando que los tres funcionan aceptablemente si son medidos por esta métrica, normalmente válida cuando la proporción de las clases de la variable dependiente es equilibrada.

Sin embargo, para el presente trabajo, tendrá mayor validez evaluar los modelos usando el área bajo la curva ROC (AUC), ya que existe un claro desbalance en las proporciones de las etiquetas a estimar (70% proyectos fracasados vs. 30% proyectos exitosos). A partir de esta métrica, solo los modelos de metainformación continúan con niveles excelentes en su performance (0.8377 para SVM y 0.7043 para MLP). Esto se debe a los resultados obtenidos a partir de sus matrices de confusión en el **subcapítulo 5.1**. El nivel del puntaje AUC para los modelos de descripciones, comparándolos con la exactitud, disminuye a un aproximado de 67% para ambos (0.6746 para SVM con TF-IDF y 0.6709 para SVM con BoW) lo cual significa que estos son modelos aún son buenos, pero no aceptables. En otras palabras, no representan modelos tan confiables para predecir el estado final de un proyecto tecnológico a partir de su descripción porque solo logran alcanzar esa cantidad de probabilidad de predicción correcta, tal como se explicó en el **subcapítulo 5.3**. sobre algunas de las deficiencias encontradas en los resultados predichos de sus matrices de confusión. Para concluir con esta métrica, el modelo de imágenes tuvo la peor performance ya que su puntaje AUC obtenido fue de 0.5010, casi 50%, lo que quiere decir que no discrimina correctamente a nivel general un proyecto fracasado de uno exitoso a partir de su imagen, pese a que en su matriz de confusión mostrada en el **subcapítulo 5.2** pronostica siempre correctamente cuando un proyecto fracasará. Este valor justamente se da por el nulo nivel de acierto de proyectos exitosos con los valores del subconjunto de prueba.

La precisión para los dos modelos de metainformación es muy alta (0.860483 para SVM y 0.930636 para MLP) ya que ambos presentan niveles muy altos de proyectos exitosos clasificados correctamente (verdaderos positivos) del total de proyectos exitosos. Casi un poco más de la mitad del umbral se encuentran las precisiones de los modelos de descripciones (0.584556 para SVM con TF-IDF y 0.532942 para SVM con BoW). No obstante, debido a su nulo poder discriminador de proyectos exitosos (0 proyectos exitosos clasificados correctamente de 810 proyectos exitosos totales en el subconjunto de prueba), el modelo de imágenes tuvo una precisión de 0.

La cuarta métrica, la sensibilidad, en contraparte con la precisión, se enfoca en el número de proyectos exitosos reales en vez de los predichos. Tanto los modelos de metainformación (0.721569 para SVM y 0.420915 para MLP) como los de descripciones (0.532349 para SVM con TF-IDF y 0.530175 para SVM con BoW) disminuyeron sus niveles en comparación con la precisión, pero mantuvieron casi las mismas proporciones que siendo evaluadas con esta última medida. El poder discriminador nulo del modelo de imágenes se mantuvo en 0.

Finalmente, y como se explicó en la teoría de métricas en el **subcapítulo 3.4**, el puntaje F1 se aplica normalmente cuando existe una diferencia considerable entre la precisión y la sensibilidad de un modelo. Por ello, actuando como un equilibrio entre ambas, para los modelos de metainformación, su puntaje F1 (0.784927 para SVM y 0.579658 para MLP) continuó respaldándolos. Para el caso de los modelos de descripciones, no es un buen evaluador ya que, tanto en la precisión como en la sensibilidad, sus niveles se mantienen constante. Y así como en los dos últimos casos, la evaluación del modelo de imágenes, ahora con F1, se mantuvo en resultado de 0.

Como conclusión final, los mejores modelos en primer lugar fueron aquellos basados en la metainformación, en especial el de Máquina de Vectores de Soporte (SVM), por sus altas performances evaluadas con todas las métricas como se explicó en los anteriores párrafos, así como por su menor tiempo de entrenamiento (16 minutos aproximado), validando así la hipótesis general. En segundo lugar, continuaron los modelos basados en las descripciones, en especial el de SVM con TF-IDF, por su mejor rendimiento y tiempo a nivel general. Ya que el modelo de imágenes solo tuvo un nivel aceptable siendo evaluado con la exactitud, y por el hecho de tener mala performance desde el punto de vista del resto de las métricas, no debe ser considerado como clasificador de estado de financiamiento de proyectos.

## 6.2. Recomendaciones

La primera recomendación dada a cualquier persona que desea seguir con este proyecto es reformular el modelo predictivo para el contenido visual que pueda ofrecer posteriormente un rendimiento entre aceptable y excelente con el objetivo de elaborar finalmente un modelo sólido alimentado por las performances de cada uno de los que lo soporta. Para ello, la primera opción es modificar las capas intermedias que este posee, desde el número de capas convolucionales hasta las de reducción, así como sus conexiones de nodos, que pueden ser desactivadas mediante capas de desactivación (*dropout*) para evitar el sobreajuste. La segunda opción es construir una arquitectura diferente de Red Neuronal Convolutacional desde cero aplicando la técnica *Data Augmentation* y posteriormente Transfer Learning al final de su capa de salida. La tercera opción para este modelo de imágenes consiste aplicar una técnica más avanzada de normalización de las características de las imágenes más allá de la normalización por lotes, para reducir el sobreajuste antes de la construcción de la nueva arquitectura del modelo, así como usar un mapa de calor que permita rastrear las características más importantes encontradas en común.

La segunda recomendación se centra en los modelos que al final de los experimentos mostraron un rendimiento más que aceptable y resultados favorecedores. Como ya se discutió en el anterior punto, y corroborando con los resultados del décimo antecedente que se tiene como base principal, el mejor modelo para entrenar la metainformación resultó el de SVM, con una ventaja pequeña sobre el de MLP. Sin embargo, partiendo de la premisa de formular un modelo global que agrupe los tres tipos de información de entrada, el modelo ganador para este caso no resulta ser el más conveniente ya que, por el hecho de trabajar finalmente con 4 variables independientes para predecir el estado de un proyecto, solo genera 4 características (los cuatro pesos finales que se obtienen al final del entrenamiento) que no se emparejan con las 128 para imágenes y texto cada uno. Ni siquiera aumentando el número de columnas a las ya mencionadas serán suficientes para cruzarlas en un posible modelo conglomerado. Es ahí donde el segundo modelo para el texto encuentra una oportunidad de poder ser seleccionado para un posterior ensamblaje de modelos si es que sus parámetros son ajustados para lograr una mejor productividad que el primero y alcanzar la cantidad mínima de características solicitadas para el cruce con las otras redes.

Como tercera recomendación planteada con la experiencia de todo el proyecto es que, sea cual sea el modelo predictivo formulado para cualquier tipo de casos de clasificación, siempre se debe asegurar que la data trabajada esté normalizada y no presente varianzas muy altas generadas por la diferencia enorme entre cada registro por casos muy particulares, como fue el caso del presente trabajo. Para ello, existen distintos métodos de normalización y estandarización de datos, entre los más empleados incluido en la actual tesis, el escalador Min-Max.

Como trabajos a futuro se plantea, además de seguir las anteriores recomendaciones, construir modelos más robustos usando datos no estructurados, como en los trabajos de los primeros antecedentes, a partir de los comentarios e interacciones de los proyectos y sus creadores con las redes sociales más populares, así como registrar las zonas más importantes de un conjunto de imágenes a través de un mapa de calor.

Finalmente, como última recomendación, aliento a toda persona a no desmotivarse por algún problema que pueda surgirle en la búsqueda de su meta principal. Como al final de este trabajo de investigación o como los casos de aquellos proyectos que no alcanzaron a ser financiados, todos los inconvenientes surgidos ayudan a tener mayor experiencia para continuar delante de alguna u otra forma hasta lograr los objetivos principales.

## BIBLIOGRAFÍA

- A Not So Random Walk. (28 de Febrero de 2019). *Backpropagation Example With Numbers Step by Step.* Obtenido de A Not So Random Walk: <https://www.anotsorandomwalk.com/backpropagation-example-with-numbers-step-by-step/>
- Alpaydin, E. (2014). *Introduction to Machine Learning* (Tercera ed.). MIT Press.
- Asociación de Emprendedores de Perú. (12 de febrero de 2018). Avances y limitaciones del emprendimiento peruano. Obtenido de <https://asep.pe/index.php/avances-limitaciones-emprendimiento-peruano/>
- Bagnato, J. I. (12 de Diciembre de 2017). *Qué es overfitting y underfitting y cómo solucionarlo.* Obtenido de Aprende Machine Learning: <https://www.aprendemachinelearning.com/que-es-overfitting-y-underfitting-y-como-solucionarlo/>
- Banafa, A. (2019). ¿Qué es el aprendizaje Profundo? *OpenMind.* Obtenido de <https://www.bbvaopenmind.com/tecnologia/mundo-digital/que-es-el-aprendizaje-profundo/>
- Beckwith, J. (2016). Predicting Success in Equity Crowdfunding. *Joseph Wharton Scholars.* Obtenido de [http://repository.upenn.edu/joseph\\_wharton\\_scholars/25](http://repository.upenn.edu/joseph_wharton_scholars/25)
- Ben Fraj, M. (5 de Enero de 2018). In Depth: Parameter tuning for SVC. *Medium.* Obtenido de <https://medium.com/all-things-ai/in-depth-parameter-tuning-for-svc-758215394769>
- Bertona, L. F. (2005). *Entrenamiento de Redes Neuronales basado en Algoritmos Evolutivos.* Tesis de grado, Universidad de Buenos Aires, Facultad de Ingeniería, Buenos Aires. Obtenido de <http://laboratorios.fi.uba.ar/lsi/bertona-tesisingenieriainformatica.pdf>
- Betancourt, G. A. (Abril de 2005). Las Máquinas de Soporte Vectorial (SVMs). *Scientia et Technica*(27). Obtenido de <https://revistas.utp.edu.co/index.php/revistaciencia/article/view/6895/4139>
- Braulio Gil, N., & Curto Díaz, J. (2015). *Customer Analytics: Mejorando la inteligencia del cliente a través de los datos.* Barcelona: Universitat Oberta de Catalunya. Obtenido de <https://docplayer.es/17897069-Customer-analytics-mejorando-la-inteligencia-del-cliente-a-traves-de-los-datos-jordi-conesa-i-caralt-coordinador-nuria-braulio-gil-josep-curto-diaz.html>
- Britos, P. V., García Martínez, R., Hossian, A., & Sierra, E. (2006). *Minería de Datos.* España: Nueva Librería.

- Calderon, P. (3 de Mayo de 2017). *Bag of Words and Tf-idf Explained*. Obtenido de Data Meets Media: <http://datameetsmedia.com/bag-of-words-tf-idf-explained/>
- Calvo, D. (13 de Julio de 2017). *Clasificación de redes neuronales artificiales*. Obtenido de Clasificación de redes neuronales artificiales: <http://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/>
- Calvo, D. (9 de Diciembre de 2018). *Definición de Red Neuronal Recurrente*. Obtenido de Red Neuronal Recurrente – RNN: <http://www.diegocalvo.es/red-neuronal-recurrente/>
- Calvo, D. (7 de Diciembre de 2018). Función de activación - Redes neuronales. España. Obtenido de <http://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/>
- Can Tayiz, B. (2019, Abril 11). Word Vectorizing and Statistical Meaning of TF-IDF. *Becoming Human: Artificial Intelligence Magazine*. Retrieved from <https://becominghuman.ai/word-vectorizing-and-statistical-meaning-of-tf-idf-d45f3142be63>
- Castelli, V. (2005). *Classification And Machine Learning Glossary*. Columbia University in the City of New York, Electrical Engineering. Obtenido de <http://www.ee.columbia.edu/~vittorio/Glossary.pdf>
- Chapel, J. (9 de Julio de 2018). AWS vs Alibaba Cloud Pricing: A Comparison of Compute Options. *Medium*. Obtenido de <https://medium.com/@jaychapel/aws-vs-alibaba-cloud-pricing-a-comparison-of-compute-options-c626d83487cc>
- Chen, S.-Y., Chen, C.-N., Chen, Y.-R., Yang, C.-W., & Lin, W.-C. (2015). Will Your Project Get the Green Light? Predicting the Success of Crowdfunding Campaigns. *Pacific Asia Conference on Information Systems (PACIS) 2015 Proceedings*. New York: Association for Information Systems AIS Electronic Library (AISel). Obtenido de <http://aisel.aisnet.org/pacis2015/79>
- Cheng, C., Tan, F., Hou, X., & Wei, Z. (2019). Success Prediction on Crowdfunding with Multimodal Deep Learning. *The Twenty-Eighth International Joint Conference on Artificial Intelligence*, (págs. 2158-2164). Macao. Obtenido de <https://www.ijcai.org/proceedings/2019/0299.pdf>
- Coding.Vision. (14 de Junio de 2013). *C# Backpropagation Tutorial (XOR)*. Obtenido de C# Backpropagation Tutorial (XOR): <https://codingvision.net/miscellaneous/c-backpropagation-tutorial-xor>
- Combinido, J. S., Mendoza, J. R., & Aborot, J. (2018). A Convolutional Neural Network Approach for Estimating Tropical Cyclone Intensity Using Satellite-based Infrared

Images. *24th International Conference on Pattern Recognition (ICPR)*. Beijing: IEEE.  
doi:10.1109/icpr.2018.8545593

- Cook, K. (27 de septiembre de 2018). Most Popular 20 Free Online Courses to Learn Deep Learning. *House of Bots*. Obtenido de <https://www.houseofbots.com/news-detail/3620-4-most-popular-20-free-online-courses-to-learn-deep-learning>
- Cyberclick. (s.f.). ¿Qué es una campaña publicitaria? *Cyberclic*. Obtenido de <https://www.cyberclick.es/publicidad/campana-publicitaria>
- DataHacker. (2018, Octubre 11). #013 C CNN VGG 16 and VGG 19. Retrieved from DataHacker: <http://datahacker.rs/deep-learning-vgg-16-vs-vgg-19/>
- Donges, N. (22 de febrero de 2018). The Random Forest Algorithm. *Towards Data Science*. Obtenido de <https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd>
- Dorofki, M., Elshafie, A. H., Jaafar, O., Karim, O. A., & Mastura, S. (2012). Comparison of Artificial Neural Network Transfer Functions Abilities to Simulate Extreme Runoff Data. En IPCBEE (Ed.), *2012 International Conference on Environment, Energy and Biotechnology*. 33, pág. 40. IACSIT Press. Obtenido de <http://www.ipcbee.com/vol33/008-ICEEB2012-B021.pdf>
- Figueroa M., G. (n.d.). Convolución y transformadas. *Revista digital Matemática*. Retrieved from <https://tecdigital.tec.ac.cr/revistamatematica/cursos-linea/EcuacionesDiferenciales/EDO-Geo/edo-cap5-geo/laplace/node7.html>
- Gandhi, R. (7 de junio de 2018). Support Vector Machine — Introduction to Machine Learning Algorithms. *Towards Data Science*. Obtenido de <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>
- Gartner. (s.f.). *Machine Learning*. Obtenido de Gartner IT Glossary: <https://www.gartner.com/it-glossary/machine-learning/>
- Gartner. (s.f.). *Predictive Modeling*. Obtenido de Gartner IT Glossary: <https://www.gartner.com/it-glossary/predictive-modeling/>
- González, L. (31 de Mayo de 2019). *Curvas ROC y Área bajo la curva (AUC)*. Obtenido de Ligdi González: Aprende todo sobre Inteligencia Artificial: <http://ligdigonzalez.com/curvas-roc-y-area-bajo-la-curva-auc-machine-learning/>
- Google Developers. (7 de septiembre de 2018). *Glosario sobre aprendizaje automático*. Obtenido de Google Developers: <https://developers.google.com/machine-learning/glossary/?hl=es-419>

- Gupta, P. (17 de mayo de 2017). Decision Trees in Machine Learning. *Towards Data Science*. Obtenido de <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052>
- Hale, J. (29 de Octubre de 2018). Best Deals in Deep Learning Cloud Providers. *Towards Data Science*. Recuperado el Junio de 2019, de [https://towardsdatascience.com/maximize-your-gpu-dollars-a9133f4e546a?fbclid=IwAR12cDltF72CE4\\_6QzZb7r\\_YVH0gtGR0QwgNyprwNVhBnqQHzlHxTNAT-qM](https://towardsdatascience.com/maximize-your-gpu-dollars-a9133f4e546a?fbclid=IwAR12cDltF72CE4_6QzZb7r_YVH0gtGR0QwgNyprwNVhBnqQHzlHxTNAT-qM)
- Hernández Sampieri, R. (2010). Capítulo 7: Concepción o elección del diseño de investigación. En *Metodología de la Investigación* (Quinta ed., págs. 118-169). México: McGrawHill.
- IArtificial.net. (2019). Matemáticas de la Regresión Logística. *Regresión Logística para Clasificación*. España. Obtenido de <https://iartificial.net/regresion-logistica-para-clasificacion/>
- IArtificial.net. (s.f.). Método del Gradiente Descendiente. *Gradiente Descendiente para aprendizaje automático*. España. Recuperado el 2019, de <https://iartificial.net/gradiente-descendiente-para-aprendizaje-automatico/>
- International Business Machines Corporation (IBM). (s.f.). *Regresión Logística*. Obtenido de IBM Knowledge Center: [https://www.ibm.com/support/knowledgecenter/es/SSLVMB\\_sub/statistics\\_mainhelp\\_ddita/spss/regression/idh\\_lreg.html](https://www.ibm.com/support/knowledgecenter/es/SSLVMB_sub/statistics_mainhelp_ddita/spss/regression/idh_lreg.html)
- Inzaugarat, E. (30 de octubre de 2018). Understanding Neural Networks: What, How and Why? *Towards Data Science*. Obtenido de <https://towardsdatascience.com/understanding-neural-networks-what-how-and-why-18ec703ebd31>
- Izco, F. (27 de Noviembre de 2018). *Base de Datos Corporativa*. Obtenido de Bookdown: [https://bookdown.org/f\\_izco/BDC-POC/metricas.html](https://bookdown.org/f_izco/BDC-POC/metricas.html)
- Jin, B., Zhao, H., Chen, E., Liu, Q., & Ge, Y. (2019). Estimating the Days to Success of Campaigns in Crowdfunding: A Deep Survival Perspective. *The 33rd AAAI Conference on Artificial Intelligence (AAAI'2019)*. Honolulu: Association for the Advancement of Artificial. Recuperado el Junio de 2019, de [http://staff.ustc.edu.cn/~cheneh/paper\\_pdf/2019/Binbin-Jin-AAAI.pdf](http://staff.ustc.edu.cn/~cheneh/paper_pdf/2019/Binbin-Jin-AAAI.pdf)
- José, I. (8 de noviembre de 2018). KNN (K-Nearest Neighbors) #1. *Towards Data Science*. Obtenido de <https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d>

- Kamath, R. S., & Kamat, R. K. (2018). Supervised Learning Model For Kickstarter Campaigns With R Mining. *International Journal of Information Technology, Modeling and Computing (IJITMC)*, 4(1). doi:10.5281/zenodo.1228716
- Kaur, H., & Gera, J. (2017). Effect of Social Media Connectivity on Success of Crowdfunding Campaigns. *Procedia Computer Science*, 122, 767-774. doi:10.1016/j.procs.2017.11.435
- Keen, B. A. (10 de Mayo de 2017). *Feature Scaling with scikit-learn*. Obtenido de Ben Alex Keen: <http://benalexkeen.com/feature-scaling-with-scikit-learn/>
- Kickstarter. (s.f.). Acerca de nosotros: Kickstarter. Obtenido de Kickstarter: <https://www.kickstarter.com/about?ref=global-footer>
- Kickstarter. (s.f.). Create something to share with others. *Intro to Kickstarter for designers, makers & technologists*. Diapositivas de PowerPoint.
- Kickstarter. (s.f.). Empieza tu proyecto. Obtenido de Kickstarter: <https://www.kickstarter.com/learn>
- Kickstarter. (s.f.). Empieza tu proyecto. Obtenido de Kickstarter: <https://www.kickstarter.com/learn?lang=es>
- Kickstarter. (s.f.). Financiamiento: Kickstarter. Obtenido de Kickstarter: <https://www.kickstarter.com/help/handbook/funding?lang=es>
- Kickstarter. (s.f.). Nuestras normas. Obtenido de Kickstarter: [https://www.kickstarter.com/rules?ref=learn\\_faq](https://www.kickstarter.com/rules?ref=learn_faq)
- Kickstarter. (s.f.). Prensa: Kickstarter. Obtenido de Kickstarter: <https://www.kickstarter.com/press?ref=hello>
- Kim, S. (16 de diciembre de 2018). Linear Regression. *Towards Data Science*. Obtenido de <https://towardsdatascience.com/regression-analysis-linear-regression-239df26a94ac>
- Kohavi, R., & Provost, F. (1998). Glossary of Terms Journal of Machine Learning. Obtenido de Stanford Artificial Intelligence Laboratory: <http://ai.stanford.edu/~ronnyk/glossary.html>
- Kraus, S., Richter, C., Brem, A., Cheng, C.-F., & Chang, M.-L. (2016). Strategies for reward-based crowdfunding campaigns. *Journal of Innovation & Knowledge*(1), 13-23. doi:10.1016/j.jik.2016.01.010
- Latorre, B. (16 de Diciembre de 2019). *The Ifs - Coding: it is child's play!* Recuperado el 17 de Diciembre de 2019, de Kickstarter: <https://www.kickstarter.com/projects/theifs/new-toy-robot-game-makes-learning-to-code-childs-play>

- Li, F.-F., Johnson, J., & Yeung, S. (16 de Abril de 2019). Convolutional Neural Networks. *CS231n: Convolutional Neural Networks for Visual Recognition*, 5, 1-80. (U. Standford, Ed.) Standford, California, Estados Unidos. Obtenido de [http://cs231n.stanford.edu/slides/2019/cs231n\\_2019\\_lecture05.pdf](http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture05.pdf)
- Li, S. (8 de julio de 2018). An End-to-End Project on Time Series Analysis and Forecasting with Python. *Towards Data Science*. Obtenido de <https://towardsdatascience.com/an-end-to-end-project-on-time-series-analysis-and-forecasting-with-python-4835e6bf050b>
- Li, Y., Rakesh, V., & Reddy, C. K. (Edits.). (2016). Project Success Prediction in Crowdfunding Environments. *WSDM' 16* (págs. 247-256). San Francisco: ACM. doi:10.1145/2835776.2835791
- Lichtig, B. (2015). Crowdfunding Success: The Short Story - Analyzing the Mix of Crowdfunded Ventures. *Wharton Research Scholars*, 121. Obtenido de [https://repository.upenn.edu/wharton\\_research\\_scholars/121/](https://repository.upenn.edu/wharton_research_scholars/121/)
- López Briega, R. E. (2 de Agosto de 2016). *Redes neuronales convolucionales con TensorFlow*. Obtenido de Raul E. Lopez Briega. Matemáticas, análisis de datos y python: <https://relopezbriega.github.io/blog/2016/08/02/redes-neuronales-convolucionales-con-tensorflow/>
- López-Golán, M., Vaca Tapia, A., Benavides García, N., & Coronado Otavalo, X. (2017). *Las campañas de crowdfunding. Su eficacia en proyectos audiovisuales en el contexto latinoamericano*. Pontificia Universidad Católica del Ecuador Sede Ibarra, Ibarra.
- Machine Learning for Artists. (s.f.). *Redes Neuronales*. Obtenido de GitHub: [https://ml4a.github.io/ml4a/es/neural\\_networks/](https://ml4a.github.io/ml4a/es/neural_networks/)
- Maimon, O., & Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook* (Primera ed.). Nueva York: Springer.
- MathWorks. (s.f.). *Máquina de vectores de soporte (SVM)*. Obtenido de Algoritmos de Machine Learning para clasificación (SVM): <https://la.mathworks.com/discovery/support-vector-machine.html>
- Merino, M. (27 de Enero de 2019). Conceptos de inteligencia artificial: qué es el aprendizaje por refuerzo. *Xataka*. Obtenido de <https://www.xataka.com/inteligencia-artificial/conceptos-inteligencia-artificial-que-aprendizaje-refuerzo>
- Microsoft. (7 de mayo de 2018). *Microsoft Docs*. Obtenido de Microsoft Naive Bayes Algorithm: <https://docs.microsoft.com/es-es/sql/analysis-services/data-mining/microsoft-naive-bayes-algorithm?view=sql-server-2017>

- Microsoft. (30 de abril de 2018). *Microsoft Docs*. Obtenido de Algoritmos de minería de datos (Analysis Services: Minería de datos): <https://docs.microsoft.com/es-mx/sql/analysis-services/data-mining/data-mining-algorithms-analysis-services-data-mining?view=sql-server-2017>
- Microsoft. (8 de enero de 2019). *Conceptos de minería de datos*. Obtenido de Microsoft Docs: <https://docs.microsoft.com/es-es/sql/analysis-services/data-mining/data-mining-concepts?view=sql-server-2017>
- Microsoft Azure. (s.f.). *Precios de Cloud Services*. Recuperado el Junio de 2019, de Microsoft Azure: <https://azure.microsoft.com/es-es/pricing/details/cloud-services/>
- Mohmad Hassim, Y. M., & Ghazali, R. (2012). *Training a Functional Link Neural Network Using an Artificial Bee Colony for Solving a Classification Problems*. Universiti Tun Hussein Onn Malaysia. Research Gate. Obtenido de [https://www.researchgate.net/publication/234005707\\_Training\\_a\\_Functional\\_Link\\_Neural\\_Network\\_Using\\_an\\_Artificial\\_Bee\\_Colonyfor\\_Solving\\_a\\_Classification\\_Problems](https://www.researchgate.net/publication/234005707_Training_a_Functional_Link_Neural_Network_Using_an_Artificial_Bee_Colonyfor_Solving_a_Classification_Problems)
- Molina Arias, M., & C, O. (Marzo de 2017). Pruebas diagnósticas con resultados continuos o políticos. Curvas ROC. *Evidencias en Pediatría*, 13(1), 1-4. Obtenido de [http://archivos.evidenciasenpediatria.es/files/41-13133-RUTA/Fundamentos\\_MBE\\_12.pdf](http://archivos.evidenciasenpediatria.es/files/41-13133-RUTA/Fundamentos_MBE_12.pdf)
- Pant, A. (22 de enero de 2019). Introduction to Logistic Regression. *Towards Data Science*. Obtenido de <https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148>
- Paperspace. (2018). *Paperspace Pricing: Virtual desktops, storage, and linux terminals*. Recuperado el Junio de 2019, de Paperspace: [https://www.paperspace.com/pricing?fbclid=IwAR0EilIW3BVjH3qVpJ9iG5FsJc4n7kYPIYHEVLt\\_3\\_uofsmYBPnze\\_fGczg](https://www.paperspace.com/pricing?fbclid=IwAR0EilIW3BVjH3qVpJ9iG5FsJc4n7kYPIYHEVLt_3_uofsmYBPnze_fGczg)
- Poole, D., Mackworth, A., & Goebel, R. (1998). Computational Intelligence: A Logical Approach. *Oxford University Press*. Recuperado el 14 de mayo de 2019
- Prabhu, R. (4 de Marzo de 2018). Understanding of Convolutional Neural Network (CNN) — Deep Learning. Medium. Obtenido de <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>

Project Management Institute. (2017). *A guide to the project management body of knowledge (PMBOK guide)* (Sexta ed.). Newtown Square, Pennsylvania, Estados Unidos: PMBOK.

Redacción Gestión. (11 de marzo de 2015). Emprendimiento en el Perú se origina más por oportunidades de negocio que por desempleo. *Diario Gestión*. Obtenido de <https://gestion.pe/economia/emprendimiento-peru-origina-oportunidad-negocio-desempleo-80578>

Redacción Gestión. (1 de agosto de 2018). Perú es el tercer país con mayor cantidad de emprendimientos en fase temprana a nivel mundial. *Diario Gestión*. Obtenido de <https://gestion.pe/economia/peru-tercer-pais-mayor-cantidad-emprendimientos-fase-temprana-nivel-mundial-240264>

Roman, V. (6 de marzo de 2019). Unsupervised Machine Learning: Clustering Analysis. *Towards Data Science*. Obtenido de <https://webcache.googleusercontent.com/search?q=cache:Mtuj4q7VUtUJ:https://towardsdatascience.com/unsupervised-machine-learning-clustering-analysis-d40f2b34ae7e+&cd=14&hl=es&ct=clnk&gl=pe>

Russell, S., & Norvig, P. (2004). *Inteligencia Artificial: Un Enfoque Moderno* (Segunda ed.). (J. M. Corchado Rodríguez, F. Martín Rubio, J. M. Cadenas Figueredo, L. D. Hernández Molinero, E. Paniagua Arís, R. Fuentetaja Pinzán, . . . R. Rizo Aldeguer, Trad.). Madrid, España: Pearson Educación, S.A. Obtenido de <https://luismejias21.files.wordpress.com/2017/09/inteligencia-artificial-un-enfoque-moderno-stuart-j-russell.pdf>

Russell, S., & Norvig, P. (2009). *Inteligencia Artificial: Un Enfoque Moderno* (Tercera ed.). Prentice Hall.

Sancho Caparrini, F. (2017). *Entrenamiento de Redes Neuronales: mejorando el Gradiente Descendiente*. Universidad de Sevilla, Departamento de Ciencias de la Computación e Inteligencia Artificial, Sevilla. Obtenido de <http://www.cs.us.es/~fsancho/?e=165>

Sancho Caparrini, F. (26 de Diciembre de 2018). *Clasificación Supervisada y No Supervisada*. Publicación, Universidad de Sevilla, Departamento de Ciencias de la Computación e Inteligencia Artificial, Sevilla. Obtenido de <http://www.cs.us.es/~fsancho/?e=77>

Sandoval, L. (s.f.). Barreras del Emprendedor ¿Por qué cuesta tanto hacerlo? *Emprender Fácil*. Obtenido de <https://www.emprender-facil.com/es/barreras-del-emprendedor/>

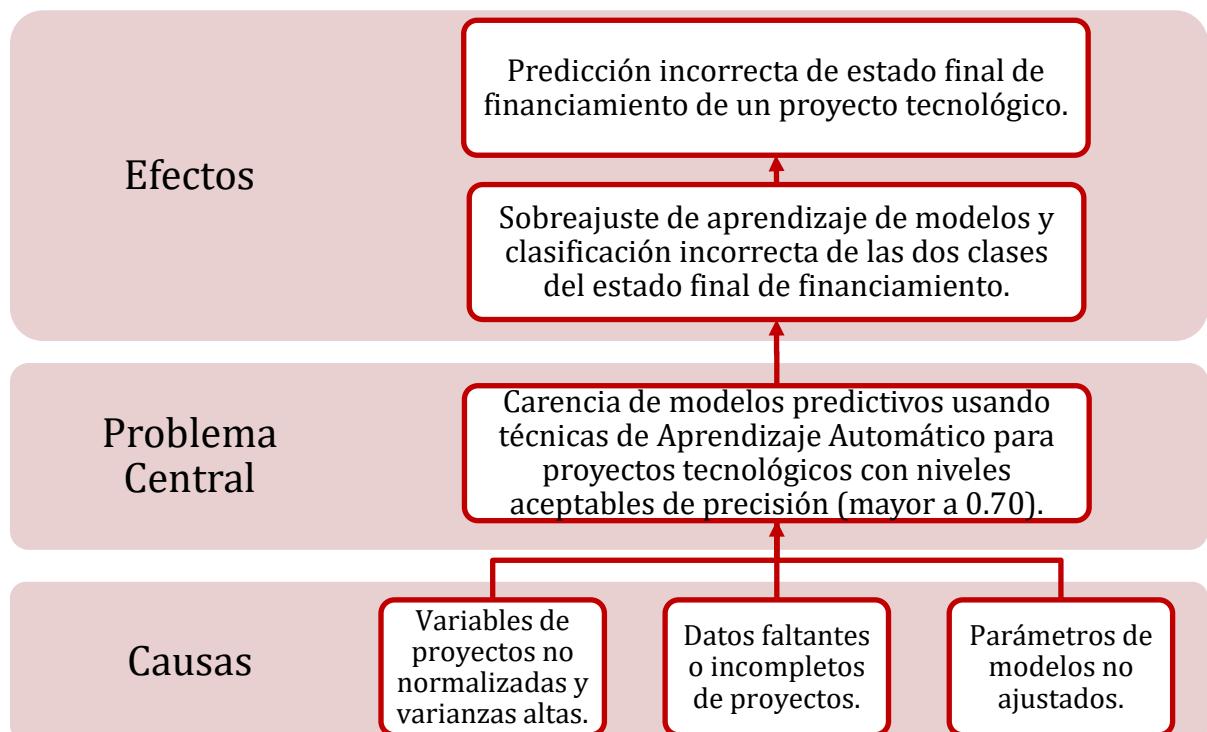
SAS Institute. (s.f.). *¿Qué es Deep Learning?* Obtenido de SAS Institute: [https://www.sas.com/es\\_pe/insights/analytics/deep-learning.html](https://www.sas.com/es_pe/insights/analytics/deep-learning.html)

- Scotiabank. (2019). *Reporte Semanal del 24 al 28 de junio del 2019*. Reporte semanal, Scotiabank, Departamento de Estudios Económicos. Recuperado el 28 de Junio de 2019, de [https://scotiabankfiles.azureedge.net/scotiabank-peru/PDFs/semanal/2019/junio/20190604sem\\_es.pdf](https://scotiabankfiles.azureedge.net/scotiabank-peru/PDFs/semanal/2019/junio/20190604sem_es.pdf)
- Shafique, U., & Qaiser, H. (noviembre de 2014). A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, 12(1), 217-222. Obtenido de <http://www.ijisr.issr-journals.org/abstract.php?article=IJISR-14-281-04>
- SitioBigData.com. (19 de Enero de 2019). Machine Learning: Selección Métricas de clasificación. Obtenido de SitioBigData.com: <http://sitiobigdata.com/2019/01/19/machine-learning-metrica-clasificacion-parte-3/#>
- SitioBigData.com. (22 de Junio de 2019). ReLU: Funciones de activación. Obtenido de <http://sitiobigdata.com/2019/06/22/relu-funciones-activacion/>
- Smola, A. J., & Schölkopf, B. (Agosto de 2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3), 199-222. Obtenido de <https://link.springer.com/article/10.1023/B:STCO.0000035301.49549.88>
- Solidaridad Latina. (s.f.). *¿Cómo funciona el crowdfunding en Latinoamérica?* Obtenido de Solidaridad Latina: <https://solidaridadlatina.com/actualizacion/como-funciona-crowdfunding-latinoamerica/>
- Sutton, R., & Barto, A. G. (2018). Finite Markov Decision Processes. En *Reinforcement Learning: An Introduction* (Segunda ed., pág. 48). Cambridge, Inglaterra: The MIT Press. Obtenido de <http://incompleteideas.net/book/RLbook2018.pdf>
- ul Hassan, M. (20 de Noviembre de 2018). *VGG16 – Convolutional Network for Classification and Detection.* Obtenido de Neurohive: <https://neurohive.io/en/popular-networks/vgg16/>
- Universo Crowdfunding. (s.f.). *¿Qué es el crowdfunding?* Obtenido de Universo Crowdfunding: <https://www.universocrowdfunding.com/que-es-el-crowdfunding/>
- Viera Balanta, V. (19 de Julio de 2013). Backpropagation explicación. Obtenido de <https://www.youtube.com/watch?v=0odQ286nsIY>
- Web Robots. (s.f.). *Acerca de nosotros: Web Robots.* Obtenido de Web Robots: <https://webrtobots.io/about-us/>
- Web Robots. (s.f.). *Kickstarter Datasets.* Recuperado el 15 de Agosto de 2019, de Web Robots: <https://webrtobots.io/kickstarter-datasets/>

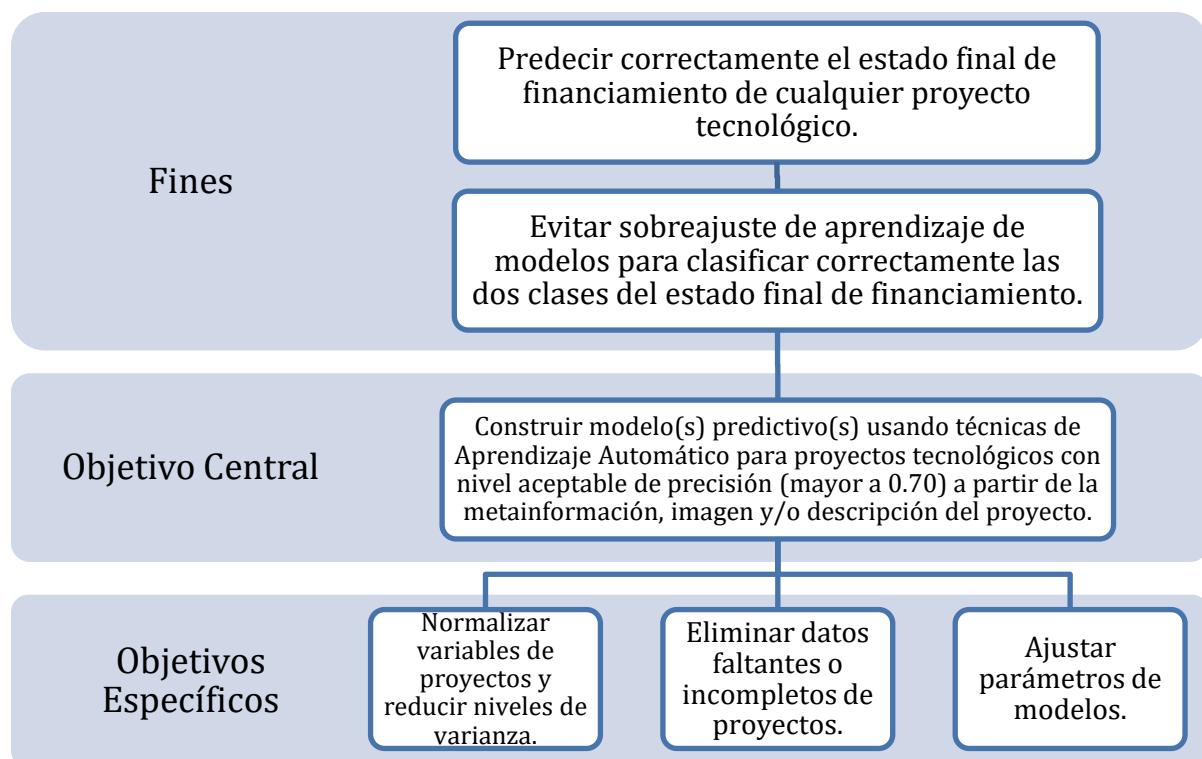
- Xuefeng, L., & Zhao, W. (2018). Using Crowdfunding in an Innovative Way: A Case Study from a Chinese Crowdfunding Platform. En IEEE (Ed.), *2018 Portland International Conference on Management of Engineering and Technology (PICMET)* (págs. 1-9). Honolulu: Portland International Conference on Management of Engineering and Technology, Inc. (PICMET). doi:10.23919/PICMET.2018.8481838
- Yu, A. (2017). *The Complete Crowdfunding Course for Kickstarter & Indiegogo*. Diapositivas de PowerPoint, Udemy.
- Yu, P.-F., Huang, F.-M., Yang, C., Liu, Y.-H., Li, Z.-Y., & Tsai, C.-H. (Edits.). (2018). Prediction of Crowdfunding Project Success with Deep Learning. *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)* (págs. 1-8). Xi'an: IEEE. doi:10.1109/ICEBE.2018.00012
- Yuan, H., Lau, R. Y., & Xu, W. (noviembre de 2016). The determinants of crowdfunding success: A semantic text analytics approach. *Decision Support Systems*, 91, 67-76. doi:10.1016/j.dss.2016.08.001
- Zambrano, J. (30 de Marzo de 2018). ¿Aprendizaje supervisado o no supervisado? Conoce sus diferencias dentro del machine learning y la automatización inteligente. *Medium*. Obtenido de <https://medium.com/@juanzambrano/aprendizaje-supervisado-o-no-supervisado-39ccf1fd6e7b>
- Zegarra García, A. (2018). *Proyecto Final del curso de Machine Learning*. Proyecto final, Universidad Peruana de Ciencias Aplicadas, Carrera de Ciencias de la Computación.
- Zhou, H. (2017). *Predicting the Success of Kickstarter Campaigns*. Proyecto final, University of California, Department of Statistics, Berkeley. Obtenido de [https://www.stat.berkeley.edu/~aldous/157/Old\\_Projects/Haochen\\_Zhou.pdf](https://www.stat.berkeley.edu/~aldous/157/Old_Projects/Haochen_Zhou.pdf)
- Zhou, M., Zhang, X., Wang, A. G., Du, Q., Qiao, Z., & Fan, W. (2015). Money Talks: A Predictive Model on Crowdfunding Success Using Project Description. *Twenty-first Americas Conference on Information Systems, Puerto Rico*. 20, págs. 259-274. Springer. doi:10.1007/s10796-016-9723-1

## ANEXOS

### ANEXO 1: Árbol de Problemas



## ANEXO 2: Árbol de Objetivos



### ANEXO 3: Glosario de términos

- ✓ Agrupamiento en clústeres (clustering): Agrupa ejemplos relacionados, particularmente durante el aprendizaje no supervisado. Una vez que todos los ejemplos están agrupados, una persona puede, de forma opcional, asignar un significado a cada clúster (Google Developers, 2018).
- ✓ Análisis de datos (data analysis): El proceso de obtener una comprensión de los datos mediante la consideración de muestras, mediciones y visualizaciones (Google Developers, 2018).
- ✓ AUC (área bajo la curva ROC): Métrica de evaluación que considera todos los umbrales de clasificación posibles. El área bajo la curva ROC es la probabilidad de que un clasificador tenga más seguridad de que un ejemplo positivo elegido al azar sea realmente positivo con respecto a que un ejemplo negativo elegido al azar sea positivo (Google Developers, 2018).
- ✓ Capa de calibración (calibration layer): Ajuste posterior a la predicción, generalmente para dar cuenta del margen de predicción. Las predicciones ajustadas y las probabilidades deben coincidir con la distribución del conjunto de etiquetas observado (Google Developers, 2018).
- ✓ Centroide (centroid): El centro de un clúster se determina mediante un algoritmo k-medios o k-mediana. Por ejemplo, si k es 3, entonces el algoritmo k-medios o k-mediana encuentra 3 centroides (Google Developers, 2018).
- ✓ Clase (class): Valor de un conjunto de valores de segmentación enumerados para una etiqueta. Por ejemplo, en un modelo de clasificación binaria que detecta spam, las dos clases son es spam y no es spam. En un modelo de clasificación de clases múltiples que identifica razas de perros, las clases serían poodle, Beagle, pug, etc (Google Developers, 2018).
- ✓ Clasificación binaria (binary classification): Tipo de tarea de predicción que da como resultado una de dos clases mutuamente exclusivas. Por ejemplo, un modelo de aprendizaje automático que evalúa mensajes de correo electrónico y da como resultado “es spam” o “no es spam” es un clasificador binario (Google Developers, 2018).
- ✓ Conjunto de datos (data set): Colección de ejemplos que coinciden con el esquema. Se representan como tablas (Kohavi & Provost, 1998).
- ✓ Convergencia (convergence): Suele referirse informalmente a un estado que se alcanza durante el entrenamiento, en el que la pérdida y la pérdida de validación cambian muy poco o nada con cada iteración después de un determinado número de iteraciones. En otras palabras, un modelo alcanza la convergencia cuando el entrenamiento adicional de los datos con los que se cuenta no mejora el modelo. En el aprendizaje profundo, los valores de

pérdida a veces permanecen constantes o casi constantes durante muchas iteraciones antes de descender finalmente, lo cual produce una falsa sensación de convergencia temporal (Google Developers, 2018).

- ✓ **Convolución (convolution):** Mezcla el filtro convolucional y la matriz de entrada para entrenar pesos. Sin convoluciones, un algoritmo de aprendizaje automático tendría que aprender un peso separado para cada celda en un tensor grande (Google Developers, 2018).
- ✓ **Datos categóricos (categorical data):** Atributos que tienen un conjunto discreto de valores posibles. Por ejemplo, considera un atributo categórico denominado house style, que tenga un conjunto discreto de tres valores posibles: Tudor, ranch, colonial. Al representar house style como datos categóricos, el modelo puede aprender los impactos de Tudor, ranch y colonial por separado en el precio de las casas. En algunas ocasiones, los valores del conjunto discreto son mutuamente exclusivos y solo se puede aplicar un valor a un ejemplo determinado (Kohavi & Provost, 1998).
- ✓ **Especificidad (specificity/True Negative Rate):** Representa el número de ítems correctamente identificados como negativos fuera del total de negativos. Se calcula mediante la siguiente fórmula (SitioBigData.com, 2019):

$$\text{Especificidad} = \frac{\text{Verdaderos negativos}}{\text{Verdaderos negativos} + \text{Falsos positivos}}$$

- ✓ **Ejemplo (example):** Fila de un conjunto de datos. Un ejemplo contiene uno o más atributos y, posiblemente, una etiqueta (Google Developers, 2018).
- ✓ **Función de activación (activation function):** Función (como ReLU o sigmoide) que incorpora la suma ponderada de todas las entradas de la capa anterior y genera un valor de resultado (generalmente no lineal) que pasa a la siguiente capa (Google Developers, 2018).
- ✓ **Lote (batch):** Conjunto de ejemplos que se usa en una iteración (es decir, una actualización del gradiente) del entrenamiento de modelos (Google Developers, 2018).
- ✓ **Modelo de clasificación (classification model):** Tipo de modelo de aprendizaje automático para distinguir entre dos o más clases discretas. Por ejemplo, un modelo de clasificación de procesamiento de lenguaje natural podría determinar si una oración de entrada está en francés, español o italiano (Castelli, 2005).
- ✓ **Modelo de referencia (baseline):** Modelo simple o heurístico que se usa como punto de partida para comparar la eficacia del desempeño de un modelo. Un modelo de referencia ayuda a los programadores de modelos a cuantificar el rendimiento mínimo esperado en un problema en particular (Google Developers, 2018).

- ✓ Ordenada al origen (bias): Una intersección o un desplazamiento del origen. En los modelos de aprendizaje automático, se hace referencia a la ordenada al origen (también conocida como el término de la ordenada al origen) como  $b$  o  $w_0$  (Google Developers, 2018). Por ejemplo, la ordenada al origen es la  $b$  en la siguiente fórmula:

$$y' = b + w_1 * x_1 + w_2 * x_2 + \dots w_n * x_n$$

- ✓ Prueba A/B (A/B testing): Forma estadística de comparar dos (o más) técnicas, generalmente con una variante nueva contra una de control. La prueba A/B tiene como objetivo determinar no solo qué técnica se desempeña mejor, sino también comprender si la diferencia tiene importancia estadística. Por lo general, la prueba A/B considera solo dos técnicas con una medición, pero se puede aplicar a un número finito de técnicas y mediciones (Google Developers, 2018).
- ✓ Punto de control (checkpoint): Datos que capturan el estado de las variables de un modelo en un momento en particular. Los puntos de control permiten exportar pesos del modelo, así como llevar a cabo el entrenamiento en varias sesiones. Los puntos de control también permiten que el entrenamiento continúe después de los errores (por ejemplo, la interrupción temporal de tareas) (Google Developers, 2018).
- ✓ Umbral de clasificación (classification threshold): Criterio de valor escalara que se aplica a la predicción de un modelo para separar la clase positiva de la negativa. Se usa al asignar resultados de regresión logística a la clasificación binaria. Por ejemplo, considera un modelo de regresión logística que determina la probabilidad de que un mensaje de correo electrónico determinado sea spam. Si el umbral de clasificación es 0.9, los valores de regresión logística por encima de 0.9 se clasifican como spam y aquellos por debajo de esa cifra se clasifican como no es spam (Google Developers, 2018).
- ✓ Valor perdido (missing value): Valor de un atributo o variable que es desconocido o no existe (Kohavi & Provost, 1998).
- ✓ Variable o atributo (feature, attribute): Una cantidad numérica o categórica que describe una instancia para ser usada como entrada (Castelli, 2005). Esta tiene un dominio definido por el tipo de atributo que denota los valores que puede tomar (Kohavi & Provost, 1998).