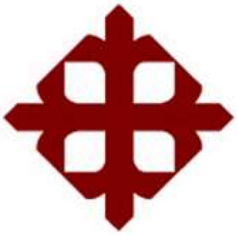


UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA:

Diseño de un Modelo de Predicción de Éxito para Proyectos Tecnológicos con Financiación en Masa Aplicando Técnicas de *Machine Learning*

AUTOR:

Yagual López Luis Manuel

Trabajo de titulación previo a la obtención del grado de
INGENIERO EN SISTEMAS COMPUTACIONALES

TUTOR:

Ing. Gallardo Posligua Vicente Adolfo, Mgs.

Guayaquil, Ecuador
11 de marzo del 2019

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

CERTIFICACIÓN

Certificamos que el presente trabajo de titulación **Diseño de un Modelo de Predicción de Éxito para Proyectos Tecnológicos con Financiación en Masa Aplicando Técnicas de *Machine Learning***, fue realizado en su totalidad por **Yagual López Luis Manuel** como requerimiento para la obtención del Título de **Ingeniero en Sistemas Computacionales**.

TUTOR

Ing. Vicente Adolfo Gallardo Posligua, Mgs.

DIRECTORA (e) DE LA CARRERA

Ing. Ana Isabel Camacho Coronel, Mgs.

Guayaquil, 11 de marzo del 2019

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Yagual López Luis Manuel**

DECLARO QUE:

El Trabajo de Titulación **Diseño de un Modelo de Predicción de Éxito para Proyectos Tecnológicos con Financiación en Masa Aplicando Técnicas de *Machine Learning*** previo a la obtención del Título de **Ingeniero en Sistemas Computacionales**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, 11 de marzo del 2019

EL AUTOR

Yagual López Luis Manuel

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

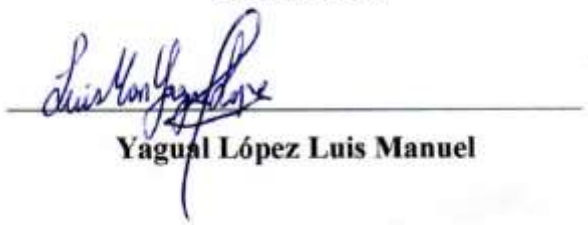
AUTORIZACIÓN

Yo, **Yagual López Luis Manuel**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación **Diseño de un Modelo de Predicción de Éxito para Proyectos Tecnológicos con Financiación en Masa Aplicando Técnicas de *Machine Learning***, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, 11 de marzo del 2019

EL AUTOR



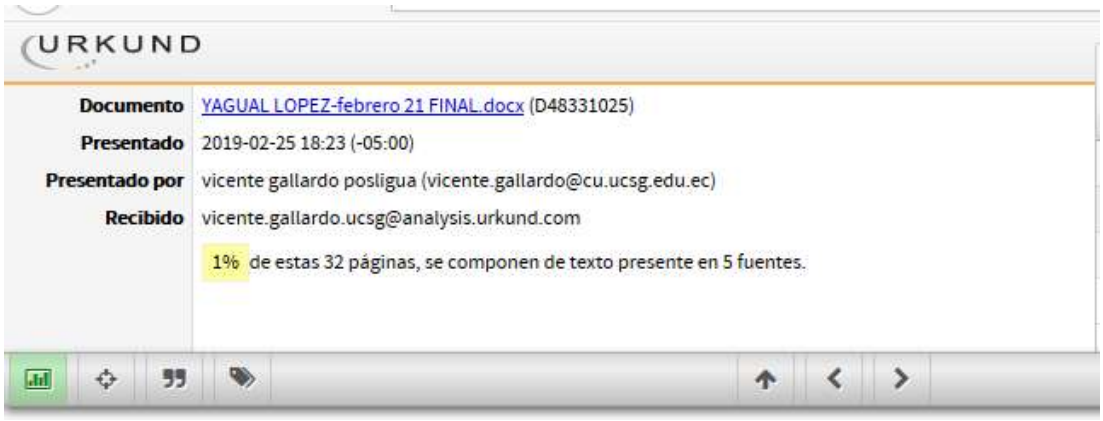
Yagual López Luis Manuel

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

REPORTE DE URKUND



AGRADECIMIENTO

Agradezco a Dios, en primer lugar, por ser guía de mi vida y por bendecirla mediante este logro y la oportunidad de estar con las personas que más amo. A mi mamá Inés, por el amor que me ha brindado cada día de mi vida y ser mi mayor ejemplo de trabajo duro y perseverancia. A mi abuela Olga, por darme amor, apoyo y mostrarme lo importante que es la disciplina y firmeza en la vida. A mi hermana Fiama por su ayuda y apoyo en este logro. A mi abuelo Manuel, por siempre apoyarme, motivarme y ser un pilar fundamental de mi vida. A mi enamorada Tamara, por motivarme a cumplir mis metas, ser siempre mi apoyo y compartir mis anhelos. A los Ing. Vicente Gallardo y Eugenio Chalén, por aportar con sus conocimientos y experiencias en el desarrollo de esta investigación.

Luis Manuel Yagual López

DEDICATORIA

Dedico este trabajo a mi mamá Inés, por su esfuerzo y amor brindado para que pudiera realizar este logro. A mis abuelos, mi hermana y mi enamorada por ser apoyos incondicionales en el camino a cumplir mis metas.

Luis Manuel Yagual López

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TRIBUNAL DE SUSTENTACIÓN

Ing. Ana Camacho Coronel, Mgs.
DIRECTORA (e) DE LA CARRERA

Ing. Cesar Salazar, Mgs.
COORDINADOR DEL ÁREA O DOCENTE DE LA CARRERA

Ing. Mario Célleri Mujica, Mgs.
OPONENTE

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE APÉNDICES

RESUMEN

Los proyectos con financiación en masa se han vuelto una tendencia en el emprendimiento a nivel global mediante el uso de plataformas tales como Kickstarter, pero estos conllevan un alto riesgo de que no logren la meta a recaudar por lo que es importante el análisis y preparación del proyecto antes de su lanzamiento; por este motivo se propone el desarrollo de un modelo de predicción de éxito para proyectos tecnológicos con financiación en masa para ayudar en la toma de decisiones previo al lanzamiento y publicación del proyecto en la plataforma Kickstarter. Para el proyecto se utilizó la investigación cualitativa, descriptiva con análisis documental como técnica de recolección de datos. Se analizaron documentos relacionados a construcción de modelos predictivos con algoritmos de aprendizaje automático y se analizaron las variables que influyen en el éxito de los proyectos en la plataforma de Kickstarter y se consolidó información de los proyectos tecnológicos de la plataforma mediante bases externas e ingreso manual. Se diseñó un flujo de trabajo, basado en prácticas generales, para el diseño del modelo predictivo y se escogieron los algoritmos de aprendizaje automático enfocados al resultado a obtener. Una vez diseñado el modelo de predicción con los algoritmos escogidos, se evaluó la exactitud y se comprobó la confiabilidad de al menos un 70% en las predicciones de éxito para los proyectos tecnológicos con financiación en masa.

Palabras clave: APRENDIZAJE AUTOMÁTICO; PROYECTOS TECNOLÓGICOS; KICKSTARTER; MODELO DE PREDICCIÓN; PYTHONJUPYTER NOTEBOOK.

INTRODUCCIÓN

En estos tiempos, gracias al gran avance del internet, se pueden encontrar páginas de financiación en masa para proyectos (crowdsourcing), donde los inversionistas y cualquier persona tienen el poder para apoyar a miles de proyectos e ideas innovadoras realizadas por emprendedores. Muchos optan por esta vía para poder financiar sus proyectos, hacerlos conocer a la sociedad y volverlos realidad. De esta forma se gana una clientela inicial y proveer del capital necesario para comenzar a trabajar y producir, pero hay un riesgo alto en especial para proyectos tecnológicos ya que éstos requieren de componentes tecnológicos como hardware y software, lo cuales pueden llegar a costar una fuerte cantidad inicial.

Es por esto por lo que se convierte en un reto brindar a los emprendedores de proyectos de tecnología la capacidad para poder analizar y obtener datos para aplicar diferentes estrategias y facilitar las tomas de decisiones en diferentes ámbitos, con el fin de poder llevar un proyecto al éxito y lograr una aceptación de los usuarios e inversores.

Gracias al avance del aprendizaje automático (Machine Learning), cuyo propósito es el diseño y estudio de las herramientas informáticas que utilizan la experiencia pasada para tomar decisiones futuras, se pueden generar modelos para analizar las variables de los proyectos tecnológicos y poder predecir si éstos van a tener éxito en la plataforma crowdsourcing.

Esta investigación pretende presentar la aplicación del Machine Learning para evaluar proyectos tecnológicos con financiamiento masivo. Los resultados de este trabajo de titulación son presentados como sigue: en el capítulo I se puede concebir la problemática a resolver, hipótesis, objetivos, justificación, alcance; el capítulo II hace referencia a ciertas teorías y principios relativos al tema en cuestión, así también algunas conceptualizaciones y mejores prácticas que requiere el modelo a diseñar; en el capítulo III está incluida la metodología de la investigación y al análisis de resultados; el capítulo IV contiene la propuesta objeto de esta investigación; cerrando con algunas conclusiones y recomendaciones.

CAPÍTULO I
EL PROBLEMA

En este capítulo se presenta el planteamiento del problema, las preguntas de investigación, los objetivos generales y específicos, la justificación y el alcance de la investigación propuesta.

1.1 Planteamiento del Problema

En la actualidad existen varios modelos de predicción de éxito que tienen diferentes metodologías para el diseño de modelos predictivos. La finalidad es contribuir con un nuevo diseño de un modelo de predicción de éxito el cual tenga su previo análisis, un flujo de trabajo y utilice técnicas de Machine Learning con el fin de que sirva de aporte, para una mejor confiabilidad al modelo predictivo y para la toma de decisión a la hora de realizar los proyectos de proyectos con financiamiento en masa.

1.2 Preguntas de Investigación

¿Se puede diseñar un modelo de predicción de éxito para proyectos tecnológicos con financiación en masa mediante técnicas de aprendizaje automático (Machine Learning)?

1.3 Objetivos de la Investigación

Los objetivos que han sido diseñados para responder a la problemática planteada son los siguientes:

1.3.1 Objetivo General

Diseñar un modelo de predicción de éxito para proyectos tecnológicos con financiación en masa mediante técnicas de Machine Learning para evaluar y seleccionar la mejor alternativa antes lanzamiento del proyecto.

1.3.2 Objetivos Específicos

- Recolectar, analizar y preparar la información de proyectos tecnológicos que usaron financiación en masa mediante uso de bases de datos externas e información de páginas web crowdsourcing
- Diseñar un modelo para predecir el éxito de proyectos tecnológicos con financiación en masa mediante herramienta Jupyter Notebook y lenguaje de programación Python.
- Evaluar y validar la confiabilidad del modelo de predicción mediante análisis e interpretación de los resultados.

1.4 Justificación

Teniendo en cuenta la gran oportunidad y ayuda que reciben los emprendedores por parte de inversionistas y el público en general a través de páginas web de financiación en masa, es necesario proveer un modelo de predicción de éxito para proyectos tecnológicos, el cual beneficia a los emprendedores y les ofrece la posibilidad de analizar y evaluar la factibilidad del proyecto antes de lanzar su campaña de crowdsourcing/crowdfunding para así asegurar una alta probabilidad de éxito y poder cautivar a la masa que aportará económicamente en la realización del proyecto.

Finalmente, este trabajo de titulación se enmarca en la línea de investigación de Inteligencia Artificial de la UCSG y Utilización de software libre de la carrera Ingeniería en Sistemas Computacionales.

1.5 Alcance

Este proyecto de investigación pretende recolectar información de base de datos externas con información relevante de proyectos tecnológicos durante el año 2017 y 2018 en la página web más popular de financiación en masa Kickstarter. Con esta data se aplicarán técnicas de Machine Learning y algoritmos de clasificación mediante el lenguaje Python con la herramienta Jupyter Notebook con el fin de diseñar el modelo de predicción de éxitos para proyectos tecnológicos que se financiaron mediante campañas crowdfunding y se evaluará el modelo para demostrar el 70% de confiabilidad.

CAPÍTULO II
MARCO TEÓRICO, CONCEPTUAL Y LEGAL

Para una mejor comprensión de la investigación es necesario una descripción de las teorías y principios de este que se han realizado en estudios previos. También se incluyen los conceptos acerca de información relevante y de las herramientas que ayudarán al desarrollo del modelo de predicción de éxito para proyectos tecnológicos con financiamientos masivo.

2.1 Teorías, principios y conceptos relativos al aprendizaje automático para proyectos con financiación en masa

En esta parte del documento se mencionan principios, investigaciones y antecedentes relacionadas a este trabajo.

2.1.1 Aprendizaje automático (Machine Learning)

En particular, se define al aprendizaje automático como un conjunto de métodos que pueden detectar automáticamente patrones en los datos, y luego usar los patrones descubiertos para predecir datos futuros, o para realizar otros tipos de toma de decisiones bajo incertidumbre (Murphy, 2012)

El aprendizaje automático se puede aplicar a una amplia gama de problemas comerciales, desde la detección de fraudes hasta la orientación al cliente y la recomendación de productos, al monitoreo industrial en tiempo real, el análisis de sentimientos y el diagnóstico médico. Puede asumir problemas que no se pueden administrar manualmente debido a la gran cantidad de datos que se deben procesar. Cuando se aplica a grandes conjuntos de datos, Machine Learning a veces puede encontrar relaciones tan sutiles que ninguna cantidad de proceso manual podría. (Brink, Richards, & Fetherolf, 2017)

El proceso de aprender de los datos, y posteriormente usar el conocimiento adquirido para informar decisiones futuras, es extremadamente eficiente. De hecho, el aprendizaje automático se está convirtiendo rápidamente en el motor que impulsa la economía moderna basada en datos (Brink et al., 2017)

Según Geron (2018) existen muchos tipos diferentes de sistemas de aprendizaje automático, por lo que es útil clasificarlos en categorías amplias basadas en:

- Si están capacitados o no con supervisión humana (supervisado, no supervisado, semi-supervisado y aprendizaje de refuerzo)
- Si pueden o no pueden aprender de forma incremental sobre la marcha (en línea frente a aprendizaje por lotes)
- Si trabajan simplemente comparando nuevos puntos de datos con puntos de datos conocidos o, en su lugar, detecten patrones en los datos de entrenamiento y construyan un modelo predictivo, de manera muy similar a como lo hacen los científicos (aprendizaje basado en ejemplos versus aprendizaje basado en modelos)

Es necesario mencionar que existen dos problemas comunes que enfrentan los algoritmos supervisados a la hora de ser entrenados con datos:

- Sobreajuste (overfitting): Se refiere a un modelo en donde los datos se han entrenado demasiado. El ajuste excesivo impacta negativamente el rendimiento del modelo y el aprendizaje con datos nuevos (Müller & Guido, 2016)
- Correlación (correlation): Ocurren cuando en el modelo se encuentran variables que al ser relacionadas con otras son el mismo valor, por ejemplo, si existen dos variables kilogramos y gramos, hay una alta correlación entre ellas ya que influyen de la misma manera, pero con diferente valor. Se encuentran también casos en el que una variable influye al 100% en el resultado final.

Existen dos tipos de aprendizaje automático supervisado que abarcan los problemas de Machine Learning, los cuales son: clasificación y regresión. En *clasificación*, el objetivo es predecir una etiqueta de clase, el cual es una opción

predefinida dentro de una lista de posibilidades. Clasificación se divide en clasificación binaria, la cual es un caso especial de distinción entre dos clases, y clasificación multiclase, la cual es una clasificación entre más de dos clases. Se puede pensar que la clasificación binaria intenta responder una pregunta de sí y no (Müller & Guido, 2016). El filtro de spam es un buen ejemplo de clasificación: es entrenado con muchos correos electrónicos junto con su clase (spam o permitido), y debe aprender a clasificar los correos electrónicos nuevos (Geron, 2018)

Para las tareas de *regresión*, el objetivo es predecir un número continuo o un número flotante en términos de programación (o un número real en términos matemáticos). Predecir el ingreso anual de una persona a partir de su educación, su edad y el lugar donde se vive es un ejemplo de una tarea de regresión (Müller & Guido, 2016)

Según Nagy (2018) la tarea de la regresión debe predecir los valores de las etiquetas (label) en función de los valores de las características (features). A menudo creamos una etiqueta cambiando los valores de una futura característica. Por ejemplo, si nos gustaría predecir los precios de las acciones en 1 mes, y creamos la etiqueta cambiando la característica del precio de las acciones 1 mes hacia el futuro, entonces:

- Para cada valor de la característica del precio de las acciones que tiene al menos 1 mes de antigüedad, deben existir datos de capacitación disponibles que muestran los datos del precio de las acciones predichos 1 mes en el futuro
- Para el último mes, los datos de predicción no están disponibles, por lo que estos valores son todos NaN (no un número)
- Debemos eliminar el último mes, porque no podemos usar estos valores para la predicción.

Mientras que la regresión se enfoca en crear un modelo que mejor se ajuste a nuestros datos para predecir el futuro, la clasificación se basa en crear un modelo que separa nuestros datos en diferentes clases (Nagy, 2018)

A continuación, se presenta una lista de los tipos de aprendizaje automático supervisado, sean de clasificación como de regresión. Son varios los algoritmos que se encuentran registrados como de clasificación:

El algoritmo **k-Nearest Neighbors** (vecinos más cercanos) o **k-NN** es posiblemente el algoritmo de aprendizaje automático más simple. k-NN es un ejemplo típico de un aprendiz perezoso. Se llama perezoso no debido a su aparente simplicidad, sino porque no aprende una función discriminativa de los datos de entrenamiento, sino que memoriza el conjunto de datos de entrenamiento (Raschka & Mirjalili, 2017). Según Raschka & Mirjalili (2017), el algoritmo KNN en sí mismo es bastante sencillo y se puede resumir mediante los siguientes pasos:

- Elegir el número de k y una métrica de distancia.
- Encontrar a los k vecinos más cercanos de la muestra que queremos clasificar.
- Asignar la etiqueta de la clase por mayoría de votos.

Según (Murphy, 2012) el algoritmo k-NN formalmente se define con la siguiente fórmula:

$$p(y = c|\mathbf{x}, \mathcal{D}, K) = \frac{1}{K} \sum_{i \in N_K(\mathbf{x}, \mathcal{D})} \mathbb{I}(y_i = c)$$

En donde NK (x, D) son los (índices de) K puntos más cercanos a **x** en D y II (e) es la función indicadora definida como: 1 si **e** es verdadero y 0 si **e** es falso. Acorde a (Murphy, 2012) se puede representar gráficamente las predicciones mediante la **Figura 1**:

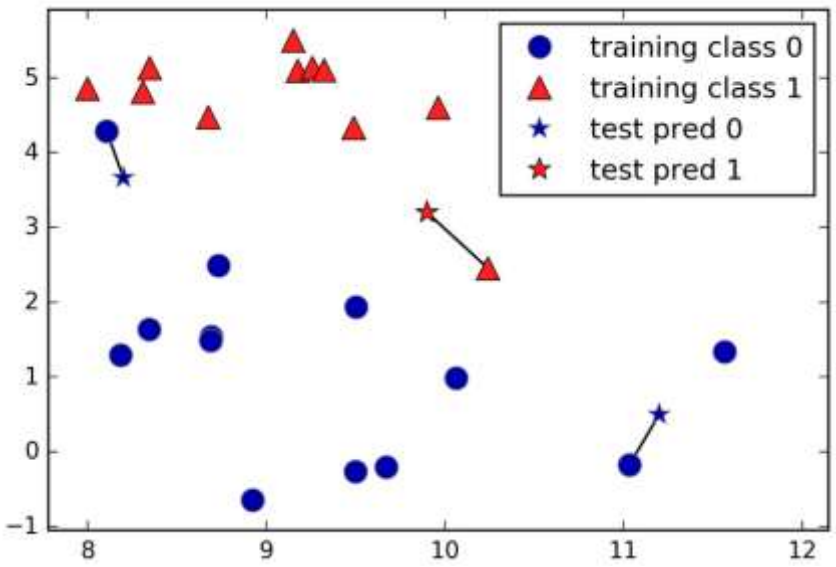


Figura 1. Predicciones hechas por el modelo K-nearest neighbor. *Nota:* Tomado de Müller & Guido (2016)

En **la Figura 1** se agregaron tres puntos nuevos de datos, mostrados como estrellas. Para cada uno de ellos, se marca el punto más cercano en el conjunto de entrenamiento. La predicción del algoritmo del vecino más cercano es la etiqueta de ese punto. Uno de los puntos fuertes de k-NN es que el modelo es muy fácil de entender y, a menudo, ofrece un rendimiento razonable sin muchos ajustes. El uso de este algoritmo es un buen método de línea de base para probar antes de considerar técnicas más avanzadas.

El algoritmo **Naive Bayes** de aprendizaje supervisado, trabaja con datos conocidos y se entrena constantemente para poder realizar cálculos probabilísticos y así poder clasificar en que pertenece el problema a predecir. Este algoritmo pertenece a la familia de clasificadores probabilísticos que computa las probabilidades de cada atributo predictivo (también llamada "feature") de los datos que pertenecen a cada clase para hacer una predicción, además de la clase más probable con la que está asociada la muestra de datos (Liu, 2017)

El algoritmo Naive Bayes utiliza el teorema de Bayes para clasificar clases y categorías. La palabra ingenuo (naive) se le dio al algoritmo porque el algoritmo asume que todos los atributos son independientes entre sí. Esto no es realmente posible, ya que cada atributo o característica en un conjunto de datos está relacionado con otro atributo, de una manera u otra (Jolly, 2018)

Según Jolly (2018) la fórmula para el teorema de Bayes es la siguiente:

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

Se puede dividir el algoritmo anterior en los siguientes componentes:

- p (h | D): Esta es la probabilidad de que se produzca una hipótesis, siempre que tengamos un conjunto de datos. Un ejemplo de esto sería la probabilidad de que se realice una transacción fraudulenta, siempre que tengamos un conjunto de datos que consistiera en transacciones fraudulentas y no fraudulentas.

- p (D | h): Esta es la probabilidad de tener los datos, dada una hipótesis. Un ejemplo de esto sería la probabilidad de tener un conjunto de datos que contenga transacciones fraudulentas.
- p (h): Esta es la probabilidad de que una hipótesis tenga lugar, en general. Un ejemplo de esto sería una declaración de que la probabilidad promedio de transacciones fraudulentas que tienen lugar en la industria móvil es del 2%.
- p (D): Esta es la probabilidad de tener los datos antes de conocer alguna hipótesis. Un ejemplo de esto sería la probabilidad de que se pueda encontrar un conjunto de datos de transacciones móviles sin saber qué queríamos hacer con él (por ejemplo, predecir transacciones móviles fraudulentas).

Murphy en su libro "Machine Learning: A probabilistic perspective", indica que la forma del algoritmo depende del tipo de característica del problema. El modelo general se representa de la siguiente manera:

$$p(\mathbf{x}|y=c,\boldsymbol{\theta}) = \prod_{j=1}^{D} p(x_j|y=c,\boldsymbol{\theta}_{jc})$$

En el caso de características de valor real, se puede usar la distribución Gaussiana; en el caso de características binarias, xj ∈ {0, 1}, se opta por la distribución de Bernoulli; y en el caso de características categóricas, xj ∈ {1, . . ., K}, se puede modelar usando la distribución multinomial.

Los modelos de Naive Bayes comparten muchas de las fortalezas y debilidades de los modelos lineales. Son muy rápidos para entrenar y predecir, y el procedimiento de entrenamiento es fácil de entender. Los modelos funcionan muy bien con datos dispersos de alta dimensión y son relativamente robustos para los parámetros. Los modelos Naive Bayes son excelentes modelos de línea de base y se usan a menudo en conjuntos de datos muy grandes, donde la capacitación, incluso en un modelo lineal, puede llevar demasiado tiempo

Otro modelo de algoritmo de este tipo es el **Árbol de decisiones** que se define mediante la partición recursiva de la entrada y la definición de un modelo local en cada región resultante. Según (Murphy, 2012) el algoritmo de clasificación de árbol de decisiones se puede representar de la siguiente manera:

$$f(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}] = \sum_{m-1}^{M} w_m \mathbb{I}(\mathbf{x} \in R_m) = \sum_{m-1}^{M} w_m \phi(\mathbf{x}; \mathbf{v}_m)$$

En donde Rm es la región m, Wm es la respuesta media en esta región, y Vm codifica la elección de la variable para la división en la ruta desde la raíz hasta la hoja m.

Los árboles de decisión son algoritmos versátiles que pueden realizar tareas tanto de clasificación como de regresión, e incluso tareas de multi-salida. Son algoritmos muy potentes, capaces de ajustar conjuntos de datos complejos (Geron, 2018)

Los árboles de decisión tienen dos ventajas sobre muchos de los algoritmos que han sido analizados hasta ahora: cualquier persona sin experiencia (al menos para árboles más pequeños) puede visualizar y comprender fácilmente el modelo resultante, y los algoritmos son completamente invariantes a la escala de los datos. Como cada función se procesa por separado y las posibles divisiones de los datos no dependen de la escala, no es necesario realizar ningún procesamiento previo como la normalización o la estandarización de las funciones para los algoritmos del árbol de decisión. En particular, los árboles de decisión funcionan bien cuando tiene características que están en escalas completamente diferentes, o una combinación de características binarias y continuas (Müller & Guido, 2016)

El principal inconveniente de los árboles de decisión es que tienden a adaptarse excesivamente y proporcionan un rendimiento de generalización deficiente.

Las **redes neuronales** son la rama más nueva de la inteligencia artificial. Las redes neuronales están inspiradas en cómo funciona el cerebro humano. La forma en que una red neuronal aprende es más compleja en comparación con otros modelos de clasificación o regresión. El modelo de red neuronal tiene muchas variables internas, y la relación entre las variables de entrada y salida puede pasar por varias capas internas. Las redes neuronales tienen mayor precisión en comparación con otros algoritmos de aprendizaje supervisado (Nagy, 2018)

Según Müller & Guido (2016), las redes neurales trabajan con varias lógicas de regresión. Las regresiones dan como salida una suma ponderada de las entradas, tal como se puede observar en la **Figura 2**:

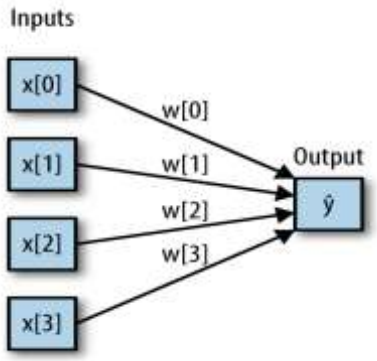


Figura 2. Visualización de una lógica de regresión. Nota: Tomado de (Müller & Guido, 2016)

En una red neuronal, este proceso de computación de sumas ponderadas se repite varias veces, primero se computan las unidades ocultas que representan un paso de procesamiento intermedio, que nuevamente se combinan utilizando sumas ponderadas para obtener el resultado final.

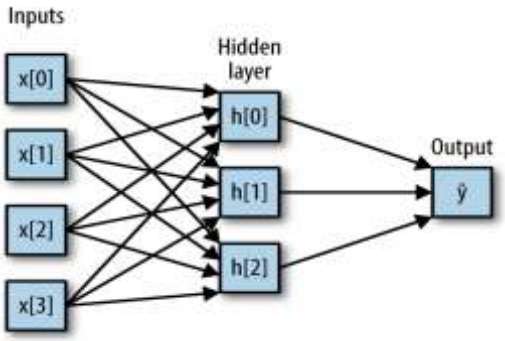


Figura 3. Visualización de una red neuronal con una capa oculta. Nota: Tomado de (Müller & Guido, 2016)

Mientras que las técnicas tradicionales de clasificación y regresión tienen sus casos de uso en inteligencia artificial, las redes neuronales artificiales generalmente son mejores para encontrar relaciones complejas entre las entradas y las salidas (Nagy, 2018)

Para los **modelos lineales de clasificación**, el límite de decisión es una función lineal de la entrada. En otras palabras, un clasificador lineal (binario) es un clasificador que separa dos clases usando una línea, un plano o un hiperplano (Müller & Guido, 2016)

Los dos algoritmos de clasificación lineal más comunes son la **regresión logística**, implementado en linear_model.LogisticRegression, y las **máquinas de**

vectores de soporte lineal (SVM lineales), implementados en svm.LinearSVC (SVC significa clasificador de vectores de soporte). A pesar de su nombre, LogisticRegression es un algoritmo de clasificación y no un algoritmo de regresión, y no debe confundirse con LinearRegression (Müller & Guido, 2016)

También son varios los algoritmos que se encuentran registrados como de regresión: La fórmula de predicción general para un **modelo lineal** de regresión tiene el siguiente aspecto:

$$\hat{y} = w[0] * x[0] + w[1] * x[1] + ... + w[p] * x[p] + b$$

Aquí, x [0] para x [p] denota las características (en este ejemplo, el número de características es p + 1) de un solo punto de datos, w y b son parámetros del modelo que se aprenden, Y es la predicción que hace el modelo. Para un conjunto de datos con una sola característica, esto es:

$$\hat{y} = w[0] * x[0] + b$$

La regresión lineal, o mínimos cuadrados ordinarios (OLS), es el método lineal más simple y clásico para la regresión. La regresión lineal encuentra los parámetros w y b que minimizan el error cuadrático medio entre las predicciones y los verdaderos objetivos de regresión, y, en el conjunto de entrenamiento. El error cuadrático medio es la suma de las diferencias cuadradas entre las predicciones y los valores verdaderos, dividida por el número de muestras. La regresión lineal no tiene parámetros, lo que es un beneficio, pero tampoco tiene manera de controlar la complejidad del modelo (Müller & Guido, 2016).

La regresión de cresta (ridge regression) y **la regresión de Lasso** son alternativas para las regresiones lineales. Según Jolly (2018) la ecuación para la regresión de cresta es la siguiente:

$$RidgeLossFunction = OLSFunction + (Alpha \times \sum Parameter1^2)$$

En la ecuación anterior, la función de pérdida de cresta es igual a la función de pérdida de mínimos cuadrados ordinarios, más el producto del cuadrado del parámetro 1 (Parameter1) de cada característica y Alpha. Alpha es un parámetro que podemos

optimizar para controlar la cantidad por la cual la función de pérdida de cresta penaliza los coeficientes, a fin de evitar el sobreajuste. Por lo tanto, la optimización de este valor de alfa proporciona el modelo óptimo que puede generalizar más allá de los datos en los que ha entrenado.

Según Jolly (2018) la ecuación para la regresión de Lasso es la siguiente:

$$LassoLossFunction = OLSFunction + (Alpha \times \sum |Parameter1|)$$

En la ecuación anterior, la función de pérdida de Lasso es igual a la función de pérdida de mínimos cuadrados ordinarios más el producto del valor absoluto de los coeficientes de cada característica y Alpha. El alfa es un parámetro que podemos optimizar para controlar la cantidad en que la función de pérdida de Lasso penaliza los coeficientes, para evitar el sobreajuste. Por lo tanto, la optimización de este valor de Alpha proporciona el modelo óptimo que generaliza mucho más allá de los datos en los que ha entrenado.

2.1.2 Mejores prácticas para aplicar técnicas de aprendizaje automático en un sistema

Según Raschka y Mirjalili (2017) el típico flujo de trabajo que comprende un proyecto de sistemas que usa aprendizaje automático es el siguiente:

- **Preproceso** (Preprocess). En esta etapa, obtenemos la data inicial. Esta data generalmente no tiene la forma que es necesaria para obtener un algoritmo de aprendizaje con buen desempeño. Por lo tanto, se prepara y se modifica esta data para dimensionarla de manera que se convierta en data preparada y el algoritmo de aprendizaje pueda hacer uso de esta para trabajar lo más optimizado posible con la data conocida y que también tenga un resultado correcto en la predicción.
- **Aprendizaje** (Learning). En esta etapa, se elige un algoritmo de aprendizaje automático con el cual trabajar y definir en el sistema.
- **Evaluación** (Evaluation). En esta etapa, después de obtener el modelo, se realizan pruebas para estimar que tan bien será el desempeño y la confiabilidad al aplicar data desconocida.

- **Predicción** (Prediction). Esta etapa arroja el resultado que predice el modelo evaluado.

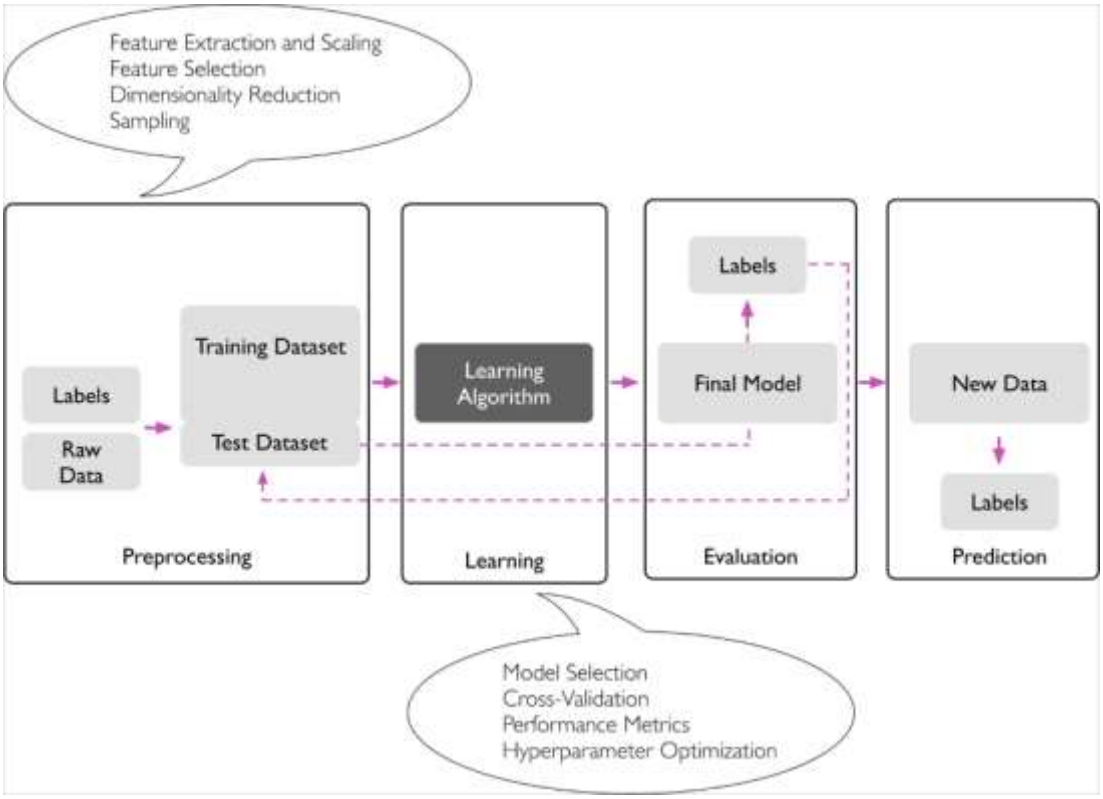


Figura 4. Flujo de trabajo del desarrollo de sistemas con Aprendizaje Automático. Nota: Tomado de Raschka y Mirjalili (2017)

2.1.3 Proyectos con financiación en masa

Financiación en masa consiste en la financiación participativa en los proyectos. La participación de personas tanto física como jurídicas, son las fuentes para la realización de proyectos de todo tipo ya sean científicos, de comercio minorista, sociales y culturales (Gracia, 2014).

Como señala Gracia (2014):

> Es posible afirmar que hoy se puede financiar, y se están financiando, todo tipo de proyectos empresariales por la vía del crowdfunding: proyectos de investigación + desarrollo, start-ups de base tecnológica o con modelos de negocio innovadores y de alto valor de crecimiento, proyectos de entretenimiento o proyectos de venta al por menor, por mencionar algunos. Los proyectos empresariales que llaman a la multitud para obtener

> financiación vienen, conforme a las fuentes que se han analizado, tanto del entorno online como off -line. Abarcan negocios de nueva creación y proyectos empresariales que buscan mejorar su explotación (p.25).

2.1.4 Proyectos de Inversión

Los proyectos de inversión son claves en la economía ya que producen viene y servicios, y además ofrecen soluciones a problemas específicos. Según Aranday (2018):

> La formulación y evaluación de proyectos de inversión tiene su origen en el plan de negocios que crea un ideólogo empresarial con el objetivo de aprovechar una oportunidad de producir bienes y servicios que satisfagan necesidades o soluciones problemas. Este plan toma forma cuando se realiza un estudio de mercado que investiga la viabilidad de la demanda por parte de la sociedad a la que se pretende dirigir el bien o servicio; así como de un estudio técnico que determina la posibilidad de producir dicho bien o servicio y la elaboración de estados financieros proforma que ordenan, numéricamente, los resultados que se obtuvieron del estudio de mercado y el estudio técnico. (p.13)

Según Flórez Uribe (2015) los pasos para la elaboración de un proyecto de inversión son:

- **Investigación preliminar**. En esta etapa de investigación preliminar, es necesario tener claridad sobre aspectos como el mercado a explotar, su demanda insatisfecha, su oferta actual y proyectada, precios o tarifas, estrategias de comercialización
- **Estudio de pre factibilidad**. En esta etapa se mejoran los aspectos identificados del mercado, los aspectos técnicos, financieros etc., elaborados en la fase anterior. Tener en cuenta que se pueden encontrar más variables para en su caso contrastar o perfeccionar las inicialmente planteadas
- **Estudio de factibilidad**. Esta etapa tiene como objetivo obtener la información precisa del plan de negocio a través del estudio del mercado, estimaciones, cronologías, ingeniería, diseño de modelos, localización etc. Además, se debe tener identificados los valores de inversión y costo, cálculos

de ingresos y demás temas financieros. También se evaluará el proyecto mediante criterios de evaluación que permitan obtener una decisión acertada a la hora de dar inicio al proyecto.

- **Diseño definitivo**. El diseño definitivo tiene como objetivo identificar y definir la estructura organizacional para la dirección del plan de negocio, así como identificar al responsable de esta.
- **Cronograma de ejecución**. En esta etapa se realiza el cronograma de todas las actividades que son parte del proyecto y además como se manejara el aspecto financiero de acuerdo con el cronograma establecido.
- **Puesta en marcha**. En la última etapa es cuando se comienza a hacer realidad lo planeado. Cabe recalcar que no todo lo establecido en las etapas previas debe ser obligatorio, ya que solo debe servir como base y guía. Siempre se pueden realizar ajustes que puedan encaminar al proyecto al éxito.

2.1.5 Startup

Según Ries (2012) "El concepto de espíritu emprendedor incluye a todo aquel que trabaje dentro de mi definición de startup: una institución humana diseñada para crear nuevos productos y servicios en unas condiciones de incertidumbre extrema". Los startups abundan y son parte de la actual economía global, debido a las innovaciones tecnológicas que van de la mano. Miles de empresas apoyan globalmente a los startups, un ejemplo es como casi 900 startups son generados al año por estudiantes del Instituto Tecnológico de Michigan MIT y que son generadoras de innovaciones y puestos de trabajo en Estados Unidos. Pero, aun así, los startups conllevan un alto riesgo a la hora de llevar el producto o servicio al mercado.

Existen varias plataformas en la web que apoyan el desarrollo de startups por parte de emprendedores a nivel mundial. Una de ellas es la más utilizada y es conocida por muchos inversores, esta plataforma web se denomina Kickstarter.

Kickstarter es una de las plataformas más populares a nivel mundial para la financiación en masa de proyectos. Kickstarter, un sitio de financiamiento de colaboración colectiva iniciado en 2009, ha sido muy popular para artistas, emprendedores con ideas ya sean tecnológicas o de proyectos sociales, los cuales necesitan financiación para llevar a cabo un proyecto.

Kickstarter funciona de una manera sencilla y es abierta a todo el público en general. La idea detrás de Kickstarter capitaliza los aspectos participativos de internet para financiar varios proyectos, aprovechando el poder de la multitud para un determinado objetivo de financiación, permitiendo a los usuarios, llamados "Patrocinadores", aportar dinero para un proyecto durante una "campaña" del proyecto a través del proceso de "capital comprometido". La campaña establece un objetivo de financiación. Si se cumple el objetivo de financiación, las promesas se convierten en contribuciones monetarias reales. Si no se cumple el objetivo de financiación, la campaña no recibe fondos. Teóricamente, cualquiera puede poner un proyecto en Kickstarter y cualquiera puede ser un patrocinador. Por ejemplo, el cantautor Alfa determino que tomaría $2500 para producir su álbum, "World Go Blue". Cuando ella comenzó su campaña de Kickstarter, estableció un objetivo de financiación de $2500. Esto significa que si los patrocinadores de esta campaña en particular invierten por lo menos$2500, el proyecto de Alfa(su álbum) recibirá la cantidad prometida, incluso si excede la meta, menos el 8%-10% que Kickstarter toma como su tarifa (Wang, 2016).

Es necesario mencionar que Kickstarter no cuenta con un medio para que usuarios puedan obtener datos de proyectos fácilmente, pero existe una plataforma web denominada WebRobots, la cual soluciona este inconveniente.

WebRobots es una plataforma web que permite a los usuarios encontrar datos de páginas web, usar herramientas para la extracción de datos o utilizar sus servicios con el fin de obtener data sets para cualquier tipo de proyectos.

2.1.6 Herramientas de desarrollo

Las herramientas de desarrollo facilitan la diseño, desarrollo y mantenimiento de aplicaciones. Estas herramientas son parte importante en la creación de aplicaciones y en diferentes actividades que procuran usar tecnología a través de los lenguajes de programación.

En el mercado existen dos lenguajes de programación para proyectos científicos que son utilizados por estudiantes, programadores, científicos, investigadores o personas con simple curiosidad: Python y R.

El lenguaje **R** es un potente lenguaje de programación y entorno para computación estadística, exploración de datos, análisis y visualización. Es gratuito, de código abierto, y tiene una comunidad fuerte y de rápido crecimiento donde los usuarios y desarrolladores comparten su experiencia y contribuyen activamente al desarrollo de paquetes, de modo que R puede resolver problemas en una amplia gama de campos (Ren, 2016)

Aunque el origen del lenguaje de programación R se remonta a 1993, su adopción general en la industria de investigación relacionada con datos ha crecido rápidamente en la última década y se ha convertido en la lengua franca de la ciencia de datos (Ren, 2016)

R, como lenguaje de programación, ha evolucionado y se ha desarrollado durante los últimos 20 años. Su objetivo es ser bastante claro para que sea fácil y flexible realizar una computación estadística completa, exploración de datos y su visualización (Ren, 2016). Sin embargo, la facilidad de uso y la flexibilidad suelen crear conflictos. Puede ser muy fácil hacer clic en algunos botones para finalizar una variedad de tareas en el análisis estadístico, pero no será flexible si necesita personalización, automatización y su trabajo debe ser reproducible. Puede ser muy flexible usar decenas de funciones para transformar datos y hacer gráficos complicados, pero no será fácil de aprender y combinar estas funciones correctamente (Ren, 2016)

Python es un lenguaje de programación interpretado, interactivo y orientado a objetos. Trabaja con diferentes estructuras de datos tales como listas, matrices, estructuras dinámicas etc. Tiene una sintaxis muy simple lo que lo hace uno de los lenguajes de programación más fáciles e intuitivos, sin embargo, es un lenguaje de programación potente y de propósito general.

Python es uno de los lenguajes de programación más populares para la ciencia de datos y por ello cuenta con una gran cantidad de bibliotecas complementarias útiles desarrolladas por su comunidad de código abierto (Raschka & Mirjalili, 2017). Si bien el rendimiento de los lenguajes interpretados, como Python, para tareas de computación intensiva es inferior a los lenguajes de programación de nivel inferior, se han desarrollado bibliotecas de extensión como NumPy y SciPy que se basan en

implementaciones de Fortran y C para operaciones rápidas y vectorizadas en matrices multidimensionales (Raschka & Mirjalili, 2017)

Una de las principales ventajas de usar Python es la capacidad de interactuar directamente con el código, utilizando un terminal u otras herramientas como Jupyter Notebook (Müller & Guido, 2016). Como consecuencia, se puede establecer algunos comparativos entre Python y R (figura 5, figura 6). Python logra una aceptación en el mercado empresarial y popularidad entre programadores debido a su gran flexibilidad y facilidad de aprendizaje. Según los lectores y miembros de la IEEE, Python ocupa el primer puesto en el ranking de lenguajes de programación, mientras que R se encuentra en el séptimo puesto.

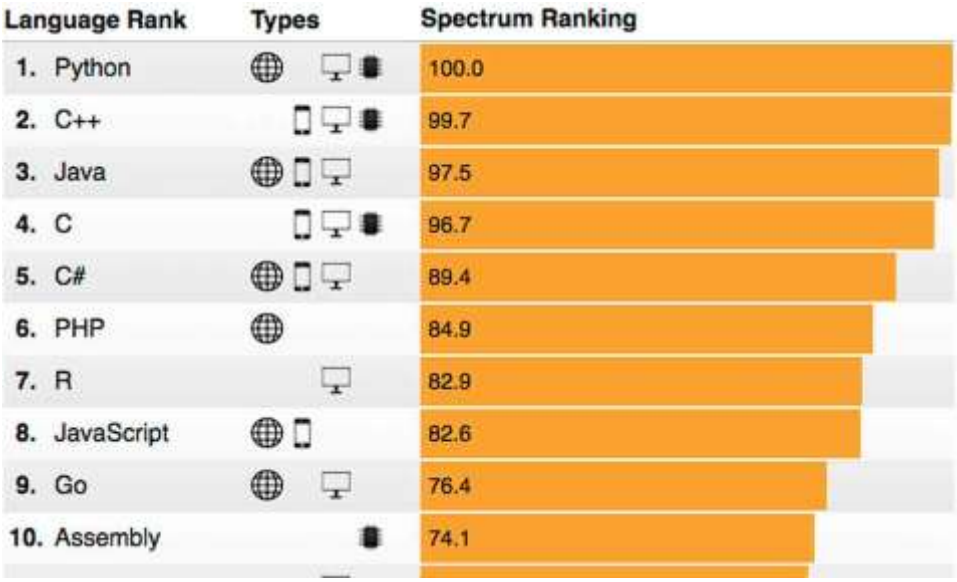


Figura 5. Ranking de Lenguajes de Programación según la IEEE. Nota: Tomado de https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages

Figura 6. Comparación entre lenguajes de programación Python y R Nota: Adaptado de www.datacamp.com

Dentro de la distribución de Python, se puede mencionar algunas distribuciones que lo complementan.

Anaconda es una distribución de Python hecha para procesamiento de datos a gran escala, análisis predictivo y computación científica. Anaconda viene con NumPy, SciPy, matplotlib, pandas, IPython, Jupyter Notebook y scikit-learn. Disponible en Mac OS, Windows y Linux, es una solución muy conveniente y es la que sugerimos para las personas sin una instalación existente de los paquetes científicos de Python (Müller & Guido, 2016)

Scikit-learn es una herramienta muy popular y la biblioteca de Python más destacada para el aprendizaje automático. Es un proyecto de código abierto, lo que significa que es de uso y distribución gratuitos, y que cualquier persona puede obtener fácilmente el código fuente. El proyecto scikit-learn se desarrolla y mejora constantemente, además tiene una comunidad de usuarios muy activa. Este contiene una serie de algoritmos de aprendizaje automático de vanguardia, así como una documentación completa sobre cada algoritmo. Es ampliamente utilizado en la industria y la academia, además de que una gran cantidad de tutoriales y fragmentos de código están disponibles en línea (Müller & Guido, 2016)

Según Toomey (2017), el producto *Jupyter* se derivó del proyecto IPython. El proyecto IPython se utilizó para proporcionar acceso interactivo en línea a Python. Con el tiempo, resultó útil para interactuar con otros lenguajes de programación, como R, de la misma manera. Separándose solo en la línea de Python, la herramienta se convirtió en Jupyter.

Jupyter se organiza en torno a algunos conceptos básicos:

- Notebook(cuaderno): Una colección de declaraciones (en un lenguaje). Por ejemplo, este podría ser un script R completo que carga datos, los analiza, genera un gráfico y registra los resultados en otros lugares.
- Cell(celda): la pieza granular más baja de un cuaderno Jupyter con la que se puede trabajar.
- Actual cell (Celda actual): la celda actual que se está editando o la (s) seleccionada (s)

- Kernel: cada cuaderno está asociado con una implementación de lenguaje específica. La parte de Jupyter que procesa el lenguaje específico involucrado se denomina núcleo (Toomey, 2017)
- Instancia (instance): Un solo objeto, observación, transacción, o registro (Brink et al., 2017)
- Etiqueta u Objetivo (label or target): El atributo numérico o categórico (etiqueta) de interés. Esta es la variable a predecir para cada nueva instancia (Brink et al., 2017)
- Características (features): Los atributos de entrada que se utilizan para predecir el objetivo. Estos también pueden ser numéricos o categóricos (Brink et al., 2017)
- Modelo (model): Un objeto matemático que describe la relación entre las características y el objetivo (Brink et al., 2017)
- Datos de entrenamiento (training data): El conjunto de instancias con un objetivo conocido que se utilizará para ajustar un modelo Machine Learning (Brink et al., 2017)

Es necesario mencionar que Python posee librerías aplicadas a el manejo de datos y machine learning, las cuales se mencionan a continuación:

NumPy es una librería fundamental para la computación científica en Python. El código de NumPy es mucho más limpio que el código Python directo y realiza las mismas tareas. Se requieren menos bucles porque las operaciones funcionan directamente en matrices y arreglos. La gran comodidad y las funciones matemáticas hacen la vida más fácil también. Los algoritmos subyacentes han pasado la prueba del tiempo y se han diseñado teniendo en cuenta el alto rendimiento (Idris, 2015)

Las matrices de NumPy se almacenan más eficientemente que una estructura de datos equivalente en Python. Array IO es significativamente más rápido también. Hay mejora en las escalas de rendimiento con el número de elementos de la matriz. Para arreglos grandes, realmente vale la pena usar NumPy. Los archivos tan grandes como varios terabytes se pueden asignar en memoria a matrices, lo que lleva a una lectura y escritura óptimas de los datos (Idris, 2015)

Pandas es una biblioteca de código abierto de Python para el análisis de datos altamente especializados. Actualmente, es el punto de referencia que todos los profesionales que utilizan el lenguaje Python deben estudiar con fines estadísticos de análisis y toma de decisiones (Nelli, 2018). De hecho, en lugar de utilizar estructuras de datos existentes integradas en Python o proporcionadas por otras bibliotecas, se desarrollaron dos nuevas estructuras de datos. Estas estructuras de datos están diseñadas para funcionar con datos relacionales o datos etiquetados, lo que le permite administrar datos con características similares a las diseñadas para bases de datos relacionales SQL y hojas de cálculo de Excel (Nelli, 2018)

Matplotlib es una biblioteca de Python especializada en el desarrollo de gráficos bidimensionales (incluidos gráficos 3D) (Nelli, 2018). Matplotlib está diseñado para reproducir tanto como sea posible un entorno similar a MATLAB en términos de vista gráfica y de forma sintáctica. Este enfoque ha sido exitoso, ya que ha podido explotar la experiencia de software (MATLAB) que ha estado en el mercado durante varios años y ahora se ha generalizado en todos los círculos técnico-científicos profesionales. Matplotlib no solo se basa en un esquema conocido y bastante familiar para la mayoría de los expertos en el campo, sino que también explota las optimizaciones que a lo largo de los años han llevado a una deducibilidad y simplicidad en su uso, lo que hace que esta biblioteca también sea una excelente opción para aquellos que se acercan a la visualización de datos por primera vez, especialmente aquellos que no tienen experiencia con aplicaciones como MATLAB o similares (Nelli, 2018)

Existen otras herramientas complementarias que ayudan en el manejo de datos, tales como:

Los *archivos CSV* son utilizado para contener grandes cantidades de data tabular, con la particularidad de que los campos del archivo son separados por coma. Estos archivos pueden ser leídos y modificados por aplicaciones tales como Microsoft Excel.

Microsoft Excel es un software producido por Microsoft que permite a los usuarios organizar, formatear y calcular datos con fórmulas utilizando un sistema de hoja de cálculo. Es una de las más populares en el mercado y utilizada en la mayoría

de las compañías afiliadas a Microsoft. Además, que cuenta con mucha facilidad de uso e interacción con el usuario.

XLTools es una extensión para Microsoft Excel que facilita las tareas de planificación y manejo de data en la aplicación. Es una extensión orientada a el aumento de la productividad y a dar una buena experiencia al usuario.

2.2 Ámbito de Aplicación

Los beneficiarios serán los emprendedores que deseen realizar un proyecto tecnológico y financiarlo mediante una plataforma de financiación en masa. Además, se beneficiarán la gente con interés en desarrollar modelos de predicciones de éxito, ya que se puede tomar esta investigación como base o guía.

CAPÍTULO III
METODOLOGÍA Y RESULTADOS

En este capítulo se describen las metodologías de investigación a emplear junto la recopilación de información requerida y el análisis de resultados, con el fin de obtener información requerida para el desarrollo de la propuesta.

3.1 Metodología de la Investigación

3.1.1 Tipo y método de investigación

El enfoque de la metodología de la investigación del presente trabajo es cualitativo descriptivo, ya que viene ajustado a los objetivos del proyecto de investigación.

El enfoque cualitativo se apoya en la recolección y resumen de datos cualitativos por medio de actividades de campo, como la realización de entrevistas, así como la observación directa y el análisis documental (Pimienta Prieto & Orden Hoz, 2012). Esta investigación tuvo un enfoque cualitativo, puesto que su finalidad es recopilar y analizar los proyectos de financiación en masa junto a sus variables para poder predecir si un proyecto de financiación en masa tendrá éxito en su recaudación de fondos y, por lo tanto, el apoyo de los usuarios e inversores, realizando el modelo mediante un algoritmo existente de Machine Learning. Se optó como método de recolección de información el análisis documental. Se analizaron los proyectos en la página web Kickstarter y en la base de datos recopilada por la página web WebRobots, con el fin de recopilar variables que se usaran para alimentar al modelo de predicción de éxito.

Dado que el propósito de esta investigación ha sido el de diseñar un modelo de predicción de éxito para proyectos tecnológicos en Kickstarter, no fue necesario determinar población ni muestra ya que se hizo una revisión documental en el sitio web para obtener la información correspondiente de los proyectos que cabe recalcar son globales y sin ningún tipo de restricción a usuarios.

Los datos recopilados con los instrumentos diseñados para ello, sirven como entrada para el modelo de predicción de éxito para proyectos tecnológicos de financiación en masa.

Se realiza también un estudio descriptivo porque se pretende detallar las variables que son más importantes en la clasificación proyectos tecnológicos financiados en masas, qué algoritmos son los más adecuados para el problema, y los mecanismos implementados en estos.

3.1.2 Técnicas e instrumentos de investigación

La técnica para recolección de información escogida fue el análisis documental ya que el objetivo fue obtener y analizar la lista de variables que se pueden visualizar en la base de datos obtenida de la página web de WebRobots, para listarlas y agregarlas a el modelo de data. También analizamos la data encontrada en los proyectos que se encuentran en la página web de Kickstarter, para poder agregar las variables adicionales a esta data y analizar el impacto que tuvieron para los proyectos que fueron exitosos, con el objetivo de escoger las variables más importantes que serán parte de nuestro modelo de predicción.

Visitando la página web www.kickstarter.com, se pueden visualizar los proyectos que son creados por los emprendedores o compañías que requieren financiamiento y podemos identificar las variables que contienen los proyectos.

Visitando la página web www.webrobots.io se pueden buscar datos de los proyectos que existen en Kickstarter. Una vez obtenido los datos se los puede recopilar, filtrar y visualizar en el software Microsoft Excel (figura 7), de modo que al final podemos tener una base de datos de los proyectos tecnológicos que se realizaron en los años 2017 y 2018. A esta base de datos se le aumentan las variables adicionales que observamos en la página de Kickstarter a cada proyecto respectivamente, para posterior análisis.

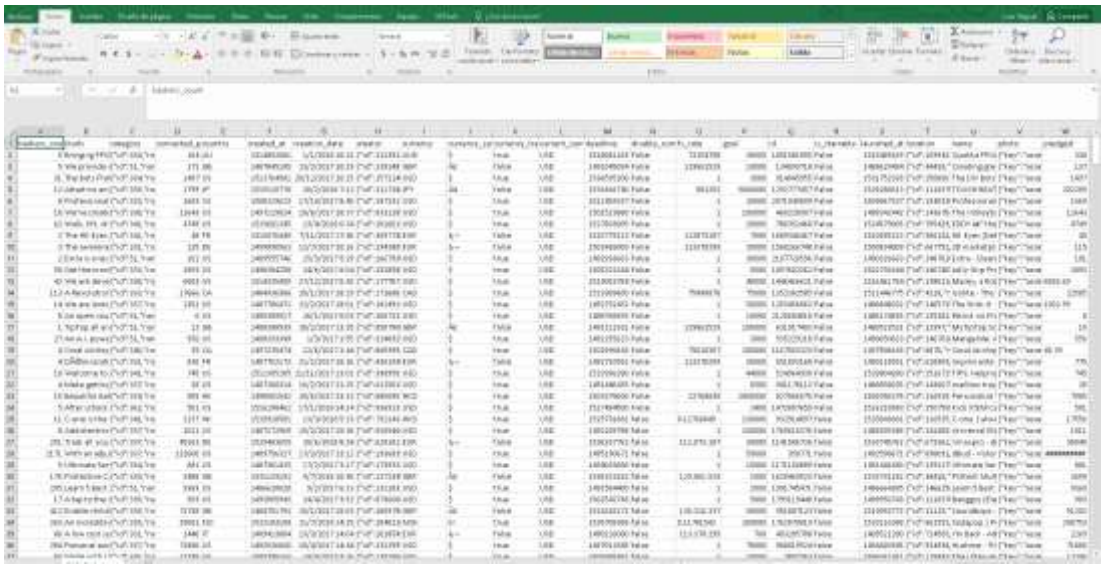


Figura 7. Ejemplo de visualización de datos de los proyectos tecnológicos en Kickstarter en Microsoft Excel.

3.2 Análisis de Resultados

En esta sección se presenta el análisis documental de los proyectos y sus variables en la página web Kickstarter, de la base de datos obtenida en la página web WebRobots y de los algoritmos que son adecuados para el problema actual.

3.2.1 Análisis de variables

Acorde al previo análisis de los datos recogidos sobre los proyectos Kickstarter en WebRobots y aplicando los respectivos filtros de tecnológicos y que se encuentren con fecha 2017-2018, se pueden visualizar que existen 511 proyectos. La base de datos ya viene con variables definidas las cuales se muestran a continuación:

- Número de patrocinadores
- Mensaje
- Categoría
- Cantidad prometida convertida
- País
- Fecha de creación
- Creador
- Meta
- ID
- ID de locación
- Nombre
- Foto
- Monto recaudado
- Perfil

- Moneda
- Símbolo de moneda
- Código de seguimiento de moneda
- Fecha de fin de recaudación
- Comunicación
- Urls
- Destacado
- Escogido por el staff
- Estado
- Fecha de cambio de estados

Por simple descarte podemos omitir las variables que no proveen valor alguno a la predicción de éxito y que solo proveen información general del proyecto o información relevante después de acabado el proyecto.

Las variables de la data con las que permaneceremos son las siguientes:

- Comunicación (disable_communication): Esta variable define si el creador del proyecto tiene o no habilitado la comunicación, ya sea por mensaje o por email, hacia los usuarios interesados.
- Meta (goal): Esta variable define el monto a recaudar puesto por el administrador del proyecto y es el que determinara si el proyecto fue o no exitoso.
- Escogido por el staff (staff_pick): Esta variable define si la página Kickstarter ha escogido al proyecto como favorita y la pondrá en primera plana para alcanzar mayor visualización. Normalmente, Kickstarter escoge a sus proyectos favoritos dependiendo del nivel de innovación de la idea y su propósito.
- Duración (duration): Esta variable define el plazo que tienen los usuarios para aportar al proyecto. Se calcula restando la fecha de fin de recaudación con la fecha de creación.
- Estado (state): Esta variable indica si el proyecto fue exitoso o no.

Acorde a la observación realizada a los proyectos en la página Kickstarter podemos definir variables adicionales que complementaran a la data para poder tener un modelo de predicción de éxito fiable.

Las variables que se encontraron fueron las siguientes:

- Video del producto
- Video Marketing
- Tipo de creador
- Primer proyecto
- Número de promesas
- Preguntas frecuentes
- Actualizaciones
- Detalle del producto
- Acerca de
- Promovido
- Contribución mínima
- Contribución máxima
- Galería de imágenes
- Limitación de envío

Para estas variables adicionales descartamos las que no tienen mayor influencia a la hora de tener éxito en un proyecto. La variable a descartar seria "Primer proyecto" (first_project), esta variable determine la experiencia del creador del proyecto. Según el siguiente análisis de los datos obtenidos, la mayoría de los proyectos tecnológicos, ya sean exitosos o fallidos, tienden a ser el primer proyecto creado debido a que no hay restricciones para crear nuevas cuentas de usuarios y esto puede llevar a no tener un historial de seguimiento fiable.

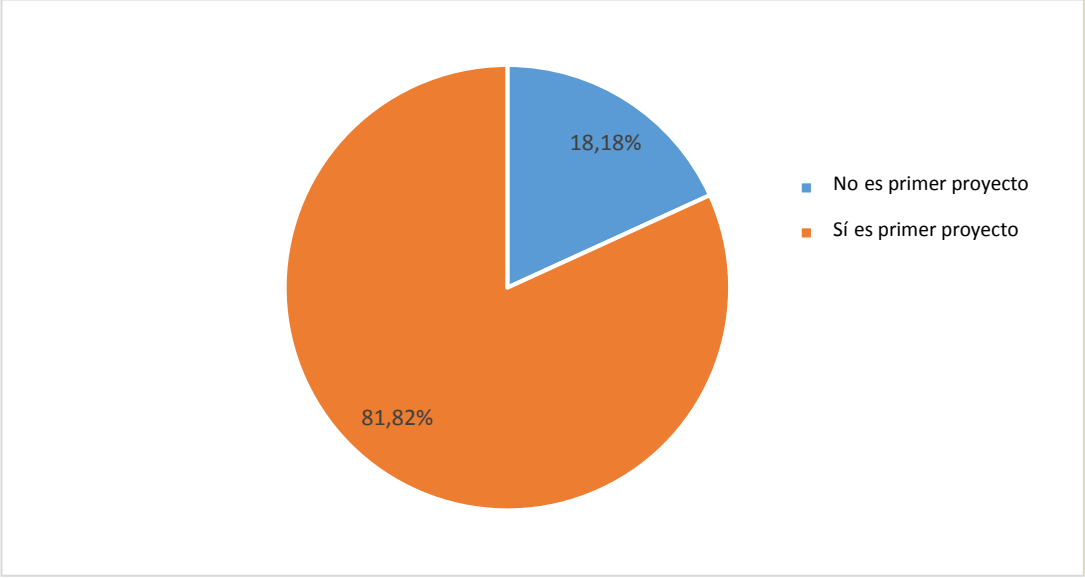


Figura 8. División porcentual de primeros proyectos tecnológicos exitosos en Kickstarter, periodo 2017-2018.

En la Figura 8, el área de color naranja indica el porcentaje de proyectos exitosos que fueron los primeros en crearse para el usuario, el cual es de un 65,71% mientras que el área color azul con un 34,29% representa a proyectos fallidos que no fueron los

primeros en crearse. Esto nos dice que la mayoría de los proyectos creados por un usuario sin experiencia pasada dentro de la plataforma, fueron exitosos.

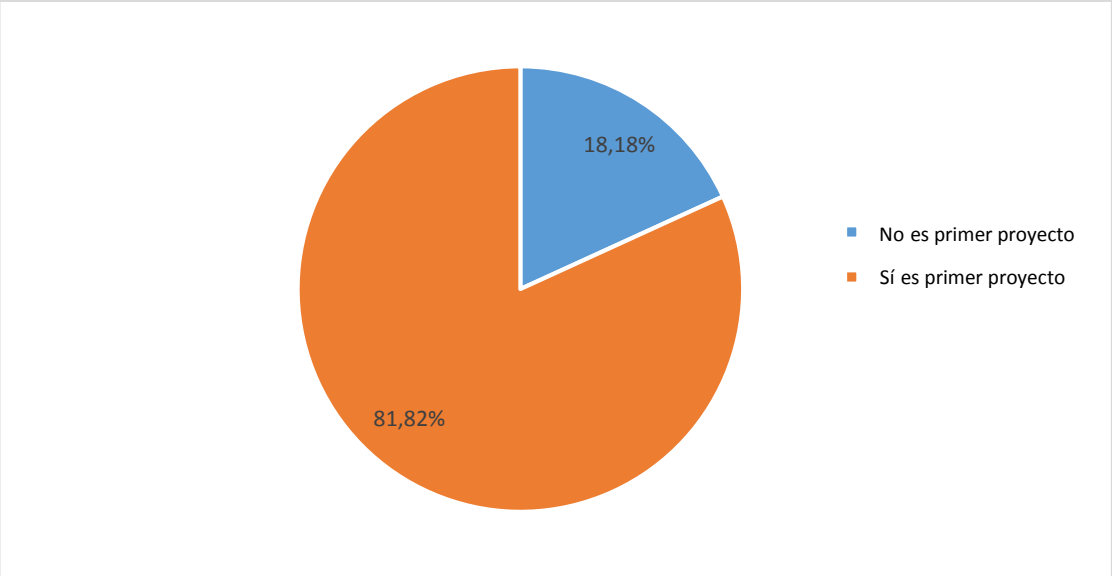


Figura 9. División porcentual de primeros proyectos tecnológicos fallidos en Kickstarter, periodo 2017-2018

En la Figura 9, el área de color naranja indica el porcentaje de proyectos fallidos que fueron los primeros en crearse para el usuario, el cual es de un 65,71% mientras que el área de color azul con un 18,18% representa a proyectos fallidos que no fueron los primeros en crearse. Esto nos dice que la mayoría de los proyectos creados por un usuario sin experiencia pasada dentro de la plataforma, fueron fallidos.

Al analizar los dos casos, se demuestra que siempre la mayoría de los proyectos serán primerizos y por lo tanto no es una variable que influya en la predicción de éxito.

Las variables que se adicionaran a la data son las siguientes:

- Video del producto (product_video): Esta variable indica si el proyecto cuenta con un video explicativo acerca del funcionamiento del producto o servicio.
- Video Marketing (marketing_video): Esta variable indica si el proyecto cuenta con un video promocional (tipo propaganda) acerca el producto o servicio.
- Tipo de creador (creator_type): Esta variable define si el tipo de creador es una persona o una compañía

- Número de promesas (number_of_promises): Esta variable indica el número de promesas que ofrece el proyecto Kickstarter a los usuarios en general.
- Preguntas frecuentes (frequented_questions): Esta variable indica si el proyecto cuenta con un apartado de preguntas frecuentes.
- Actualizaciones (updates): Esta variable indica si el proyecto cuenta con un apartado de actualizaciones.
- Detalle del proyecto (product_explanation): Esta variable indica si el proyecto fue detallado a los usuarios mediante un apartado de texto e imágenes.
- Acerca de (about_us_explanation): Esta variable indica si el proyecto cuenta con un apartado de "Acerca de" en el cual se puede tener un trasfondo de quien o quienes son responsables del proyecto.
- Promovido (promoted): Esta variable indica si el proyecto está promovido por alguna organización tercera a Kickstarter y a el creador del proyecto.
- Contribución mínima (mínimum_contribution): Esta variable indica el monto de contribución mínima que tiene el proyecto.
- Contribución máxima (máximum_contribution): Esta variable indica el monto de contribución máxima que tiene el proyecto.
- Galería de imágenes (picture_gallery): Esta variable indica si el proyecto cuenta con una galería de imágenes acerca del producto o servicio.
- Limitación de envío (limited_courier): Esta variable indica si el proyecto tiene limitaciones de países en cuanto a envíos de sus promesas a los usuarios.

3.2.2 Consolidación de información

Como resultado de la eliminación de variables que no se utilizarían para alimentar el modelo de predicción y la adición de variables adicionales, podemos obtener una base de datos robusta acerca de los proyectos tecnológicos que se realizaron el periodo 2017 y 2018 (Figura 10).

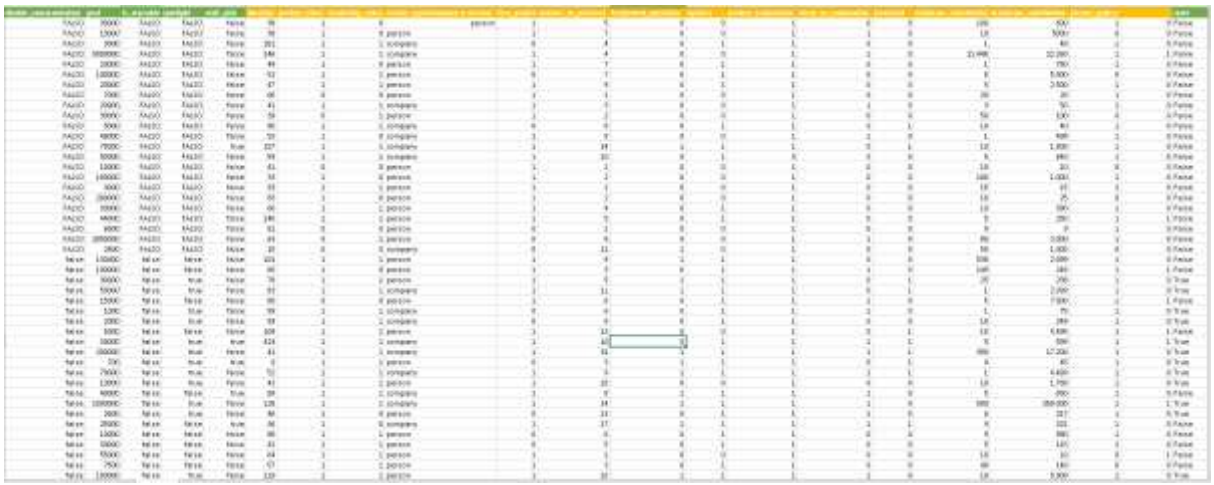


Figura 10. Consolidación de variables de proyectos tecnológicos en Kickstarter

3.2.3 Elección de lenguaje de programación

Después del análisis y comparación entre los dos lenguajes de programación, se escoge a Python como herramienta para esta investigación debido a la facilidad de uso, entendimiento y rapidez para la codificación. Gracias al uso de paquetes y la herramienta Jupyter Notebook, se pueden realizar análisis de datos, aplicar algoritmos y técnicas de Machine Learning para diseñarlos y visualizarlos en un solo lugar.

3.2.4 Elección de algoritmos

Según los libros "Python machine learning: machine learning and deep learning with Python, scikit-learn, and TensorFlow" de Raschka, S., & Mirjalili, V. y "Introduction to machine learning with Python: a guide for data scientists" de Müller, A. C., & Guido, S. se puede determinar cuáles son los algoritmos óptimos para este problema en específico mediante el análisis de cuanta data se procesará, el tipo de problema y el resultado que se quiere obtener. Además, se puede elaborar el flujo de trabajo adecuado para el diseño y evaluación del modelo de predicción de éxito.

- Algoritmo Naive Bayes: Se usará este algoritmo debido a la rapidez en su entrenamiento, su facilidad de aprendizaje y su funcionamiento adecuado para datos robustos con muchas variables. El clasificador que se implementará con scikit-learn será GaussianNB, debido a que este puede aplicarse a cualquier dato continuo mientras que los demás son usados para clasificaciones de datos de texto.

- Algoritmo Árbol de decisiones: Se usará este algoritmo para comparación con el anterior debido a su facilidad para visualizaciones de análisis de datos y su adaptabilidad a la escala de datos. El clasificador que se implementará con scikit-learn será RandomForestClassifier.

3.2.5 Flujo de trabajo

Si bien no existe ninguna norma a seguir para la elaboración de modelos de predicción, existen mejores prácticas que ayudaran al entendimiento y para lograr el mejor desempeño a la hora de aplicar las técnicas de Machine Learning. Como resultado del análisis de estos, se procede a unificar el flujo de trabajo (Figura 11) y se definen cinco procesos como base:

- Definición del planteamiento de la solución
- Preparación de la data
- Selección del algoritmo
- Entrenamiento del modelo
- Validación del modelo

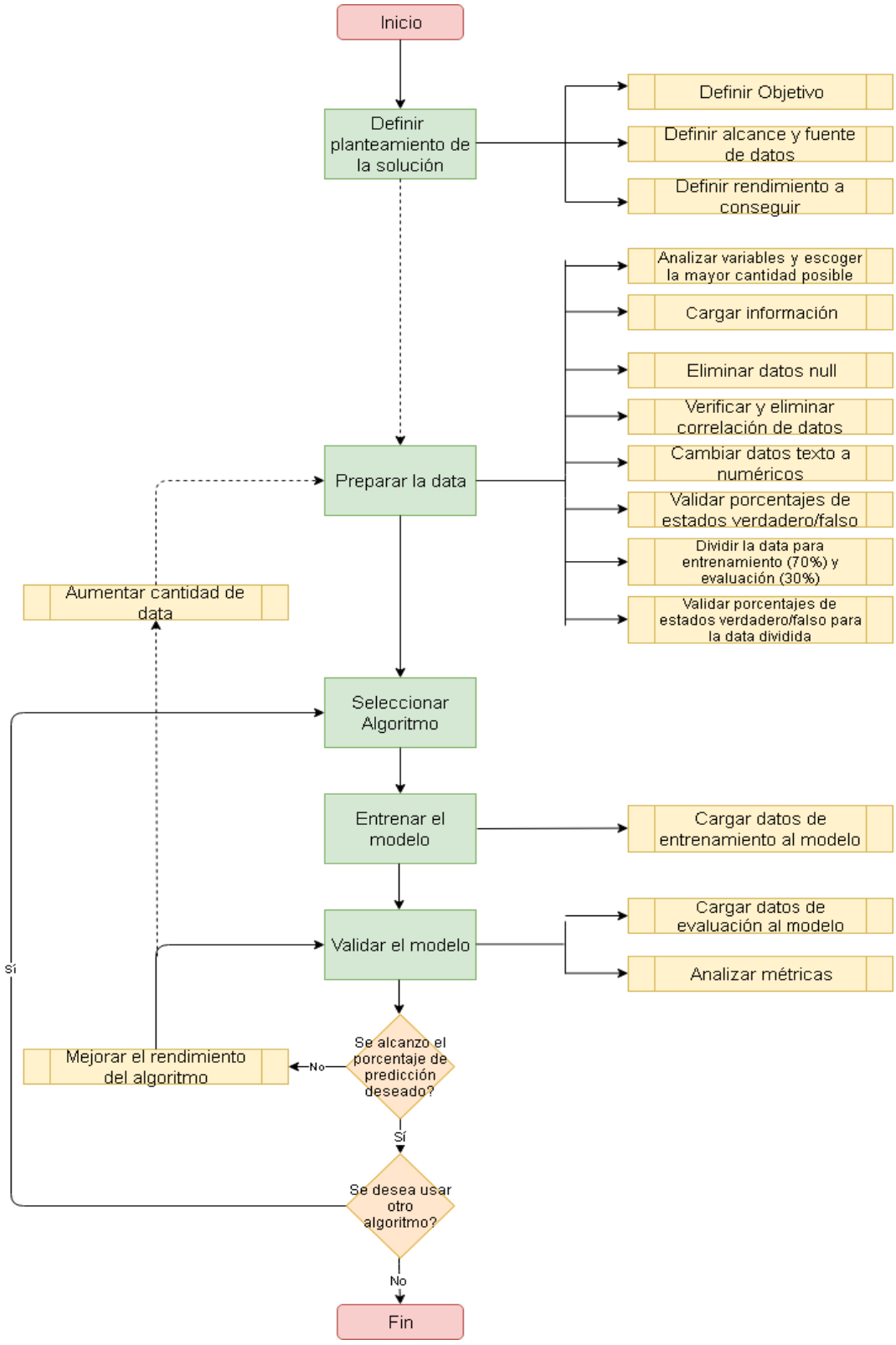


Figura 11. Diagrama de flujo de trabajo para diseñar modelo de predicción mediante Machine Learning

CAPÍTULO IV
PROPUESTA

Con el fin de atender las necesidades señaladas de obtener un modelo de predicción de éxito confiable para proyectos tecnológicos de financiamiento en masa, se ha diseñado el modelo preparando la data de los proyectos de la página web Kickstarter y armando el modelo según las variables adecuadas que fueron analizadas en el capítulo anterior. Una vez preparada la data, esta servirá para alimentar los algoritmos escogidos, con el fin de analizar los resultados de predicción del modelo y optimizar según sea necesario, todo mediante la aplicación Jupyter Notebook.

4.1 Objetivo

Diseñar un modelo de predicción de éxito confiable y robusto mediante un flujo de trabajo establecido, con el fin de que se pueda facilitar la toma de decisiones al crear proyectos con financiamiento en masa.

4.2 Descripción

El diseño del modelo de predicción de éxito para proyectos tecnológicos con financiamiento en masa está compuesto por diferentes fases, depende de varias herramientas tecnológicas y de un proceso minucioso que se mencionan a continuación:

4.2.1 Fases de la solución

Definición el planteamiento de la solución: En esta etapa se define el objetivo, el alcance y origen de datos y el rendimiento del modelo a conseguir.

Preparación de la data: En esta etapa la data es filtrada y evaluada junto a los atributos y clases para asegurar que todo este correcto. El propósito de dichas actividades es para que el modelo, junto a un algoritmo, pueda ser alimentado por dicha data.

Selección del algoritmo: En esta etapa se selecciona el algoritmo que más se apegue a el problema en cuestión.

Entrenamiento del modelo: En esta etapa se cargan los datos al modelo para ser alimentado y se dividen en dos: entrenamiento y evaluación. Normalmente el 70% de la data se toma para entrenamiento y el 30% para evaluación.

Validación del modelo: En esta etapa se visualizan las métricas y se analizan los resultados del modelo.

4.2.2 Herramientas tecnológicas

Python: Este lenguaje de programación fue elegido para el desarrollo del diseño del modelo debido a su facilidad de lectura y escritura, su comunidad y sus librerías que facilitan la aplicación de técnicas de aprendizaje automático. Estas y otras características fueron mencionadas en el capítulo 3.2.1.3.

Anaconda: Esta distribución de Python fue escogida ya que contiene todas las librerías y herramientas a usar para computación científica. Anaconda se usó en esta investigación para tener acceso a las librerías y herramientas a usar sin necesidad de varias instalaciones.

Numpy: Esta librería se utiliza para tener una mayor facilidad a la hora de crear los arreglos y vectores con el código Python.

Pandas: Esta librería se utiliza para poder manejar los datos iniciales, que se encuentran en un archivo, en estructuras de datos diseñadas para funcionar con datos etiquetados y permite una administración de datos similar a la de una base de datos.

Matplotlib: Esta librería se utiliza para poder construir y visualizar gráficos de datos y poder analizar con mayor facilidad los resultados.

Scikit-learn: Esta herramienta contiene una biblioteca completa de algoritmos de aprendizaje automático. Se usó para poder aplicar los algoritmos Naive Bayes y el árbol de decisiones Random Forest.

Jupyter Notebook: Esta herramienta sirve como intérprete de código para poder realizar todo el proceso del diseño y aplicar el uso de las librerías y bibliotecas previamente mencionadas.

WebRobots: Esta página web proporciona los datos de todos los proyectos que se encuentran en la plataforma de financiamiento en masa Kickstarter.

Microsoft Excel: Esta herramienta sirve para contener todos los datos iniciales obtenidos en WebRobots y los datos posteriormente agregados, luego del análisis de variables realizado en el capítulo 3.2.1.1, para agruparlos en un solo lugar.

XLTools: Esta extensión de Microsoft Excel se utiliza para la exportación de los datos a un archivo CSV con el fin de ser usado por la librería Pandas.

4.2.3 Requerimientos de hardware y software

Para instalar la distribución de Anaconda y utilizar la herramienta de Jupyter Notebook se necesita de los siguientes requerimientos de hardware y software:

- Sistema operativo: Windows 7 o mayor, 64 bits macOS 10.10+, o Linux Ubuntu, RedHat, CentOS 6+.
- Arquitectura de sistemas: Windows 64 bits x86, 32 bit x86; MacOS-64bit x86; Linux 64it x86, 64 bit Power8/9
- Minimo 5GB en disco para descargar e instalación.

4.2.4 Proceso

La propuesta del diseño del modelo para la predicción de proyectos tecnológicos con financiamiento en masa está compuesta de:

- Elección de página de financiamiento en masa: Se elige la página Kickstarter debido a que es la más popular de financiamiento en masa. Esta actividad es parte de la primera etapa del flujo de trabajo.
- Análisis de variables de proyectos tecnológicos en Kickstarter: Se realizo un análisis detallado de todas las variables que pueden influir en el éxito de un proyecto en Kickstarter. Esta actividad es parte de la primera etapa del flujo de trabajo.
- Flujo de Trabajo para el diseño del modelo: Este flujo se realizó luego del análisis documental de mejores prácticos y procesos para diseñar un modelo.

- Selección del algoritmo: Los algoritmos se escogieron de acuerdo a el tipo de investigación (algoritmos de Machine Learning supervisados binarios), el resultado que queremos obtener y las ventajas que ofrecen.

El proceso para diseñar el modelo de predicción de éxito para proyectos de financiamiento en masa sigue paso a paso el flujo de trabajo determinado en el capítulo 3.2.1.5, el cual es detallado a continuación:

4.2.4.1 Planteamiento de la solución

En esta etapa se definen tres variables para la investigación: objetivo, alcance y origen de datos.

- Objetivo: Predicción de éxito de proyectos
- Origen de datos: Proyectos tecnológicos de Kickstarter
- Alcance: Proyectos Kickstarter entre 2017-2018
- Rendimiento del modelo mínimo 70%.

4.2.4.2 Preparación de la data

Analizar variables y escoger la mayor cantidad posible

En esta etapa se analizaron todas las variables que comprendían los proyectos de financiación en masa en la página web de Kickstarter y de una lista se escogieron las más convenientes a usar, tal como se explica en el capítulo 3.2.1.1.

Carga de data

Se consolidó la información desde WebRobots, el cual contiene los datos de los proyectos Kickstarter, mediante la aplicación Microsoft Excel y se procedió a filtrar la data por tipo de proyecto tecnológico y que se encuentren en el rango de años 2017-2018.

La variable “category” es la que contiene el tipo del proyecto y se filtró si contiene el texto “technology”.

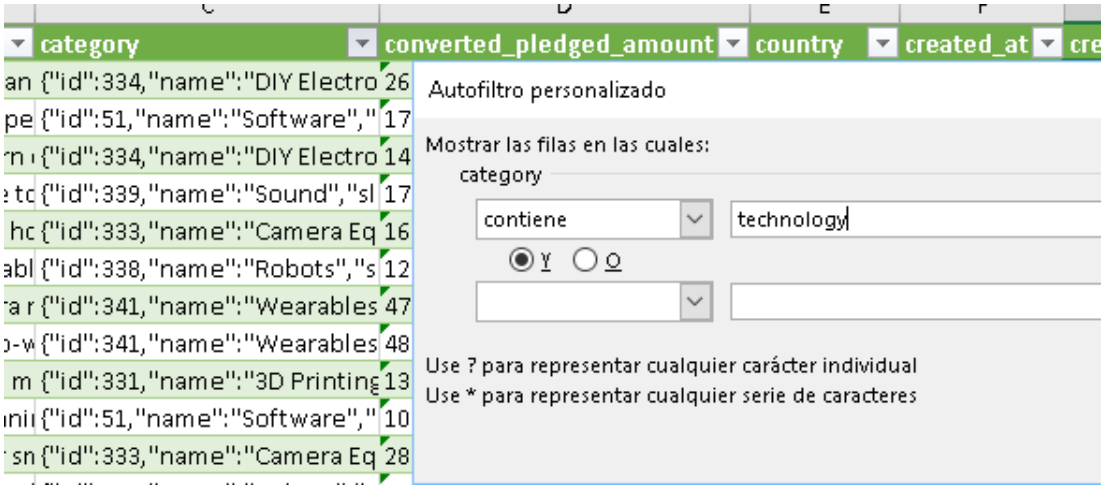


Figura 12. Captura de pantalla de filtro por categoría "technology" en campo "category"

La variable de "created_at" la cual contiene la fecha de creación del proyecto viene por defecto en formato UNIX, por lo que se la convirtió a formato de Fecha. (Ver Apéndice A). Una vez formateada correctamente se pudo filtrar por los años 2017 y 2018.

Para asegurar que el modelo de predicción sea confiable y robusto, se analizó una a una las variables como se determina en el capítulo 3.2.1.1 y se agregó y elimino según correspondían. Una vez determinadas las nuevas variables se procedió a visitar uno por uno los proyectos en Kickstarter para poblar la información. Cabe mencionar que las nuevas variables eran de carácter booleano y solo podían tener como dato verdadero o falso, cuando son verdaderos se ingresó el valor 1 y cuando son falsos se ingresó el valor 0.

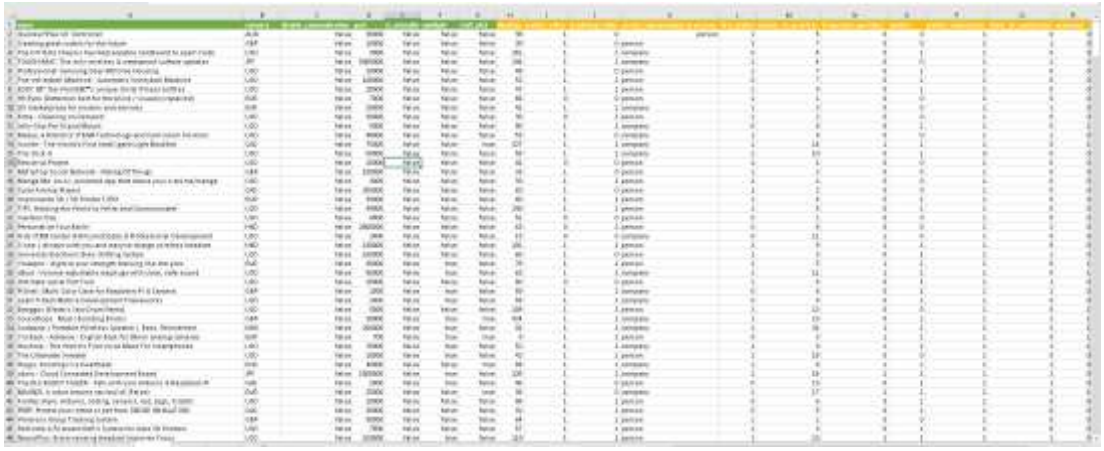


Figura 13. Captura de pantalla proyectos tecnológicos consolidado

Una vez terminado el llenado de información, se procedió a exportar el archivo Excel a formato CSV mediante el plugin XLTools. Después se procedió a abrir nuestro

proyecto en la herramienta Jupyter para cargar nuestro archivo CSV (Ver Apéndice A).

Eliminación de datos null y variables no analizables

Una vez cargado nuestro archivo, se procedió a verificar si existen campos que no contengan ningún valor y estén en blanco CSV (Ver Apéndice A). La herramienta presentó el resultado "False" que indica que no existen datos nulos, por lo que se procedió al siguiente paso, que fue el de eliminar las variables no analizables tales como nombre del proyecto.

Verificar y eliminar correlación de datos.

Se verificó si existían correlación de datos en la herramienta generando un gráfico mediante la función de correlación (Ver Apéndice A), en donde se comparan una a una las variables desde el eje x al eje y. Se marca de color amarillo cuando existe correlación en la comparación de dos variables, lo cual se espera que pase cuando se compara una variable a sí misma, mas no con otra diferente. En la figura 15 de correlación se determinó que la variable "Elegida por el Staff" tenía una alta correlación con la variable "Estado" la cual indica si el proyecto fue exitoso o no, por lo que se eliminó la variable "Elegida por el Staff" (Ver Apéndice A).

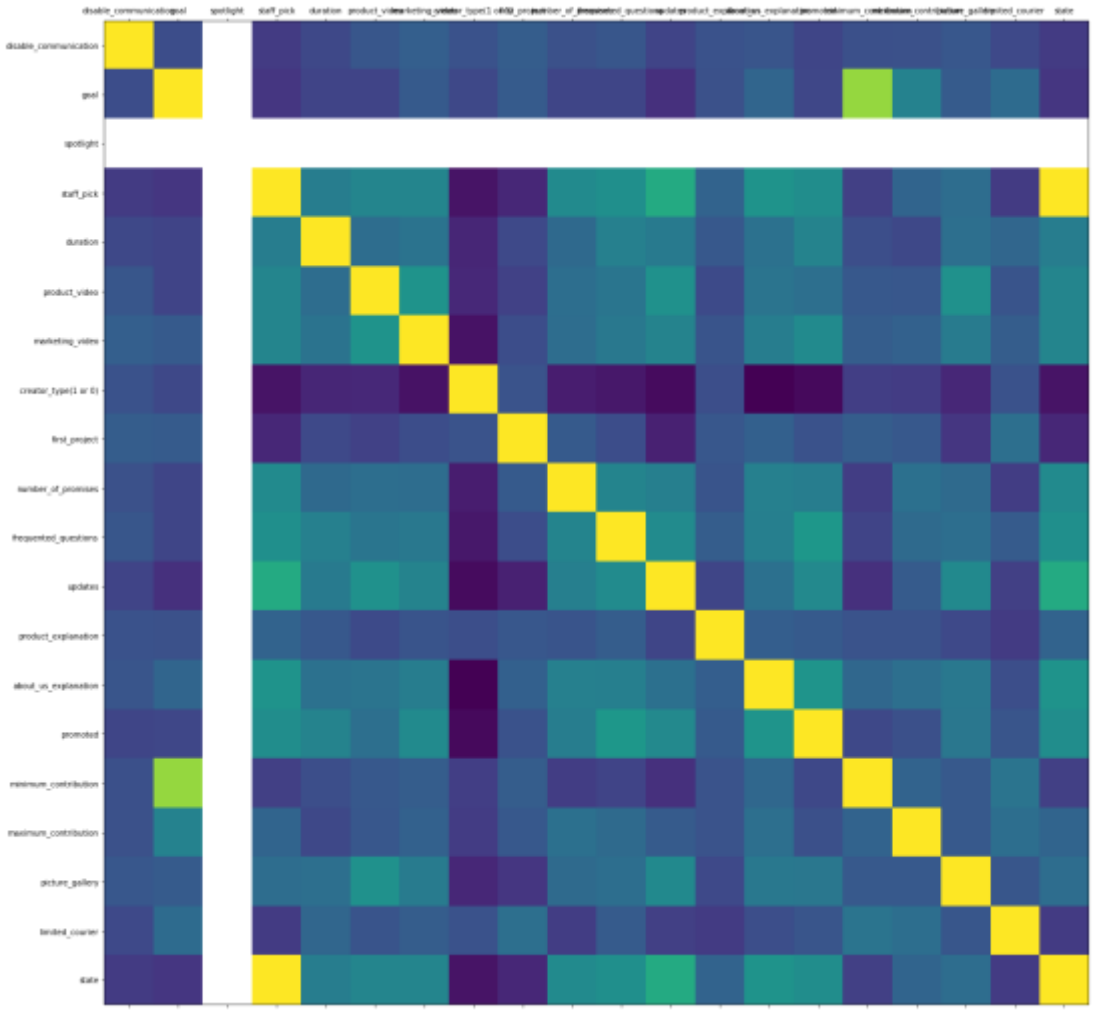


Figura 15. Gráfico de correlación de variables en Jupyter Notebook

4.2.4.3 Selección del algoritmo

En esta etapa se analizaron todos los algoritmos de tipo supervisado y de clasificación que podían servir de ayuda a la hora de realizar el modelo de predicción. Se escogieron dos algoritmos: Naive Bayes y Árbol de decisiones.

4.2.4.4 Entrenamiento del modelo

Cargar datos de entrenamiento al modelo. En esta etapa se alimentó a los algoritmos Naive Bayes y Árbol de decisiones (Random Forest) con la data destinada a entrenamiento (Ver Apéndice B).

4.2.4.5 Validación del modelo

Cargar datos de evaluación al modelo. En esta etapa se alimentó a los algoritmos Naive Bayes y Árbol de decisiones (Random Forest) con la data destinada a evaluación para que puedan ser predichas por el modelo (Ver Apéndice B).

Analizar métricas. En esta etapa se mostró los resultados del rendimiento y exactitud de los algoritmos a la hora de predecir los datos de evaluación. También se graficó el árbol de decisiones y se determinó cuáles eran las variables con más peso para el éxito de un proyecto. (Ver Apéndice B).

4.3 Resultados

Los resultados serán presentados a continuación para describir el rendimiento del modelo de predicción comparando los dos algoritmos escogidos. Además, se presentará el análisis del árbol de decisiones de las variables con más peso para determinar los puntos más importantes que se deben tomar en cuenta a la hora de usar financiación en masa para los emprendimientos.

En la figura 16 se contempla los resultados de la ejecución del modelo predictivo con el algoritmo Naive Bayes alimentado por la data de evaluación. A pesar de que la data cuenta con una variedad de variables para hacer al algoritmo más confiable a la hora de predecir los proyectos tecnológicos en Kickstarter, se verificó que la precisión del modelo solo nos da un porcentaje de exactitud del 56% para la data de entrenamiento y 48% con data de evaluación. Esto significa que el modelo de predicción no es muy confiable debido a su bajo valor de exactitud. El modelo de entrenamiento no ha tenido la suficiente cantidad de data esperada por el algoritmo Naive Bayes para poder realizar una predicción con mayor exactitud.

Prediciendo training data

```
[22]: nb_predict_train = nb_model.predict(X_train)


      from sklearn import metrics

      #training metrics
      print("Accuracy: {0:.4f}".format(metrics.accuracy_score(y_train, nb_predict_train)))
```

```
Accuracy: 0.5690
```

Prediciendo test data

```
[23]: # predict values using the testing data
      nb_predict_test = nb_model.predict(X_test)

      from sklearn import metrics

      #training metrics
      print("Accuracy: {0:.4f}".format(metrics.accuracy_score(y_test, nb_predict_test)))
```

```
Accuracy: 0.4800
```

Figura 16. Gráfico del resultado de exactitud del modelo usando el algoritmo Naive Bayes.

Mientras que en la figura 17 se contempla los resultados de la ejecución del modelo predictivo con el algoritmo Random Forest alimentado por la data de evaluación. Al usar este algoritmo obtenemos una exactitud de predicción del 100% para datos de entrenamiento y del 70% para datos de evaluación, cumpliendo con el objetivo de obtener un modelo confiable. Esto se debe a que el algoritmo de árbol de decisiones requiere de menor data de entrenamiento para lograr el mínimo porcentaje de exactitud esperado y por lo cual es la mejor opción si no se planea aumentar la data de entrenamiento.

Prediciendo Training data

```
[26]: rf_predict_train = rf_model.predict(X_train)
      # training metrics
      print("Accuracy: {0:.4f}".format(metrics.accuracy_score(y_train, rf_predict_train)))
```

```
Accuracy: 1.0000
```

Prediciendo Test Data

```
[27]: rf_predict_test= rf_model.predict(X_test)

      print("Accuracy: {0:.4f}".format(metrics.accuracy_score(y_test, rf_predict_test)))
```

```
Accuracy: 0.7067
```

Figura 17. Gráfico del resultado de exactitud del modelo usando el algoritmo Random Forest

En la figura 18 se contempla el resultado de la ejecución del código para poder visualizar en un ranking cuales son las variables con más peso a la hora de influir en el éxito de un proyecto. Por lo cual se puede observar que las variables de mayor peso son:

- Número de promesas
- Contribución mínima
- Actualizaciones
- Meta
- Duración

```
31]: important_features = RandomForestClassifier()
     important_features = important_features.fit(X_train,y_train)
     importances = important_features.feature_importances_
     std = np.std([tree.feature_importances_ for tree in important_features.estimators_], axis=0)
     ind = np.argsort(importances)[::-1]

     print("Ranking:")

     for f in range(X_train.shape[1]):
         print("%d. variable %d (%f)" % (f +1, ind[f], importances[ind[f]]))

     Ranking:
     1. variable 6 (0.195159)
     2. variable 13 (0.134024)
     3. variable 14 (0.107363)
     4. variable 1 (0.102959)
     5. variable 8 (0.094685)
     6. variable 3 (0.091091)
     7. variable 7 (0.065807)
     8. variable 5 (0.047635)
     9. variable 10 (0.045176)
     10. variable 4 (0.041142)
     11. variable 11 (0.030336)
     12. variable 15 (0.021669)
     13. variable 12 (0.019194)
     14. variable 0 (0.002277)
     15. variable 9 (0.001482)

:  ['0-disable_communication', '1-goal

:  ['0-disable_communication',
     '1-goal',
     '2-spotlight',
     '3-duration',
     '4-product_video',
     '5-marketing_video',
     '6-number_of_promises',
     '7-frequented_questions',
     '8-updates',
     '9-product_explanation',
     '10-about_us_explanation',
     '11-promoted',
     '12-maximum_contribution',
     '13-minimum_contribution',
     '14-picture_gallery',
     '15-limited_courier']
```

Figura 18. Gráfico del ranking de peso de las variables usando algoritmo Random Forest.

En la figura 19, se puede observar el árbol de decisiones generado por la herramienta y poder analizar los caminos que toma el modelo al aprender mediante el algoritmo Random Forest y las diferentes variables proveídas.

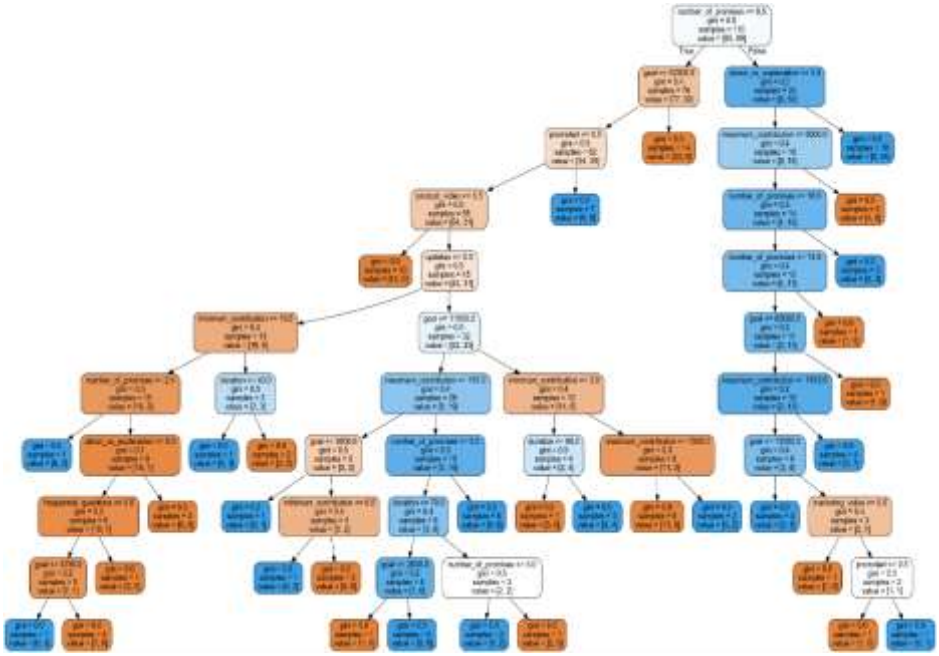


Figura 19. Gráfico del árbol de decisiones generado por el algoritmo Random Forest

Al lograr un valor mayor al 70% en la exactitud del modelo a la hora de predecir el éxito de los proyectos con financiación en masa y determinar cuáles son las variables más influyentes y con mayor peso a la hora del éxito de un proyecto, se puede ahora usar el modelo de predicción para mediante ingreso de datos y así nos pueda indicar si nuestro proyecto tendrá un alto porcentaje de éxito o de fracaso. (Ver Apéndice B).

CONCLUSIONES

Al término del proyecto se estableció que es posible diseñar un modelo de predicción para proyectos tecnológicos con financiamiento en masa mediante técnicas de Machine Learning.

Una vez diseñado el modelo es posible obtener una exactitud de predicción mayor o igual al 70% para el modelo de predicción de éxito de proyectos tecnológicos con financiación en masa mediante el algoritmo Random Forest, si se tiene limitado el número de data con el cual entrenar al algoritmo.

A través del algoritmo Random Forest se puede determinar que las variables más importantes a la hora de influir en el éxito de un proyecto tecnológico con financiamiento en masa son las siguientes: meta, mínima contribución, el número de promesas, actualizaciones y duración.

RECOMENDACIONES

Realizar el modelo de predicción de proyectos tecnológicos aumentado el número de años y así obtener más data de entrenamiento para asegurar una exactitud de mínimo 70% con el algoritmo Naive Bayes.

Realizar el modelo de predicción de proyectos aumentando el alcance de datos de otras páginas de financiamiento, para así poder tener más flexibilidad y beneficios a la hora de crear emprendimientos basados en los resultados del modelo.

Realizar un aplicativo el cual haga uso de la base del diseño de modelo de predicciones para proyectos tecnológicos con financiamiento en masa, previamente diseñado usando el flujo de trabajo determinado en esta investigación, con el objetivo de tener una mejor experiencia frente a un usuario sin conocimientos técnicos.

REFERENCIAS BIBLIOGRÁFICAS

Aranday, F. R. (2018). *Formulación y evaluación de proyectos de inversión.: Una propuesta metodológica*. IMCP. Recuperado de https://books.google.com.ec/books?id=Qs9XDwAAQBAJ

Brink, H., Richards, J. W., & Fetherolf, M. (2017). *Real-world machine learning*. Shelter Island: Manning.

Flórez Uribe, J. A. (2015). *Proyectos de inversión para las PYME (3a. ed.).* Bogotá: Ecoe Ediciones. Recuperado de http://public.eblib.com/choice/publicfullrecord.aspx?p=4422269

Geron, A. (2018). *HANDS-ON MACHINE LEARNING WITH SCIKIT-LEARN, KERAS, AND TENSORFLOW: concepts, tools, and ... techniques to build intelligent systems.* Place of publication not identified: O'REILLY MEDIA.

Gracia, L. (2014). Presente y futuro del crowdfunding como fuente de financiación de proyectos empresariales. *Revista Española de Capital Riesgo*. Recuperado de https://docplayer.es/2583662-Presente-y-futuro-del-crowdfunding-como-fuente-de-financiacion-de-proyectos-empresariales.html

Idris, I. (2015). *NumPy: Beginner's Guide - Third Edition*. Recuperado de http://sbiproxy.uqac.ca/login?url=http://international.scholarvox.com/book/88853039

Jolly, K. (2018). *Machine Learning with scikit-learn quick start guide: classification, regression, and clustering techniques in Python*. Recuperado de http://proquest.safaribooksonline.com/?fpi=9781789343700

Liu, Y. (2017). *Python machine learning by example: easy-to-follow examples that get you up and running with machine learning*. Birmingham Mumbai: Packt Publishing.

Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists* (First edition). Sebastopol, CA: O'Reilly Media, Inc.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. Cambridge, MA: MIT Press.

Nagy, Z. (2018). *BEGINNING AI, MACHINE LEARNING AND PYTHON: get started with the development of real-world ... applications that are powered by the latest ai adv.* Place of publication not identified: PACKT Publishing Limited.

Nelli, F. (2018). *Python Data Analytics: With Pandas, NumPy, and Matplotlib*.

Pimienta Prieto, J. H., & Orden Hoz, A. de la. (2012). *Metodología de la investigación.* Distrito Federal: Pearson Educación.

Raschka, S., & Mirjalili, V. (2017). *Python machine learning: machine learning and deep learning with Python, scikit-learn, and TensorFlow* (Second edition). Birmingham Mumbai: Packt Publishing Ltd.

Ren, K. (2016). *Learning R Programming*. Recuperado de http://sbiproxy.uqac.ca/login?url=http://international.scholarvox.com/book/88843441

Toomey, D. (2017). *Jupyter for Data Science: exploratory analysis, statistical modeling, machine learning, and data visualization with Jupyter*. Birmingham Mumbai: Packt.

Wang. (2016). The Promise of Kickstarter: Extents to Which Social Networks Enable Alternate Avenues of Economic Viability for Independent Musicians Through Crowdfunding. *Social Media + Society*. Recuperado de https://journals.sagepub.com/doi/pdf/10.1177/2056305116662394

APÉNDICES

Apéndice A. Manual de Implementación

Objetivo

Ayudar al usuario a instalar las herramientas necesarias para el desarrollo de esta investigación.

Descargar Anaconda

Anaconda es una distribución de Python muy popular y contiene todos las herramientas y librerías que necesitamos.

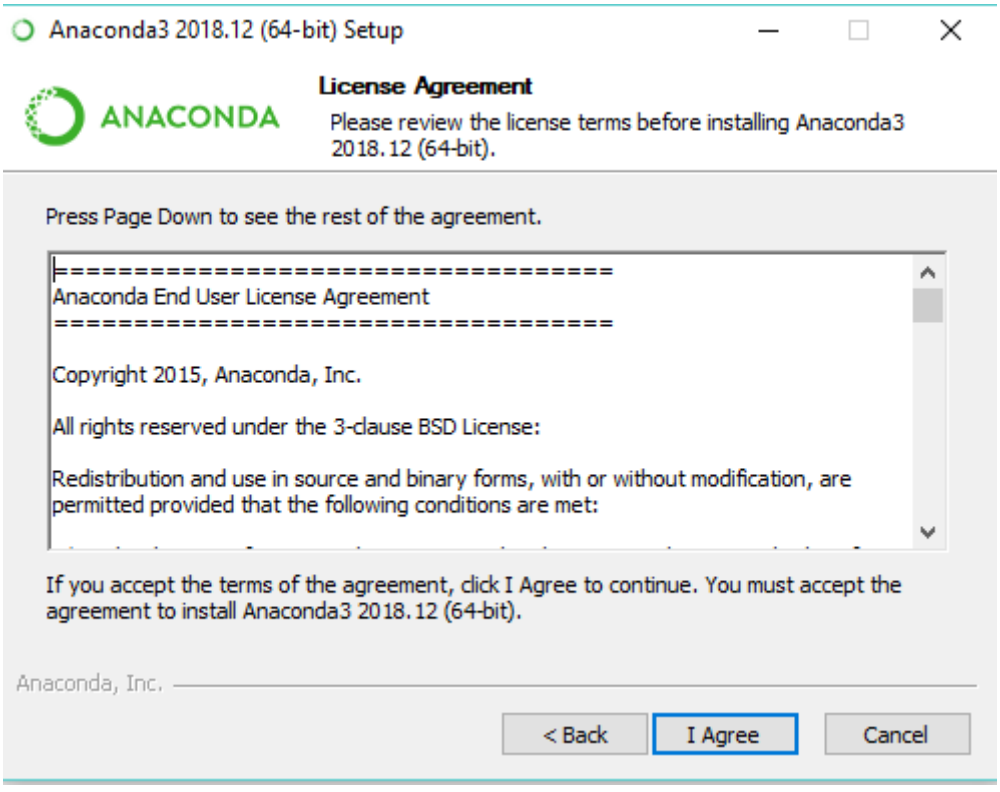
Para descargar la distribución Anaconda ingresar al siguiente link: https://www.anaconda.com/distribution/#download-section

En la sección de sistemas operativos elegir Windows y descargar la última versión de Python disponible dando clic en la opción “Download”

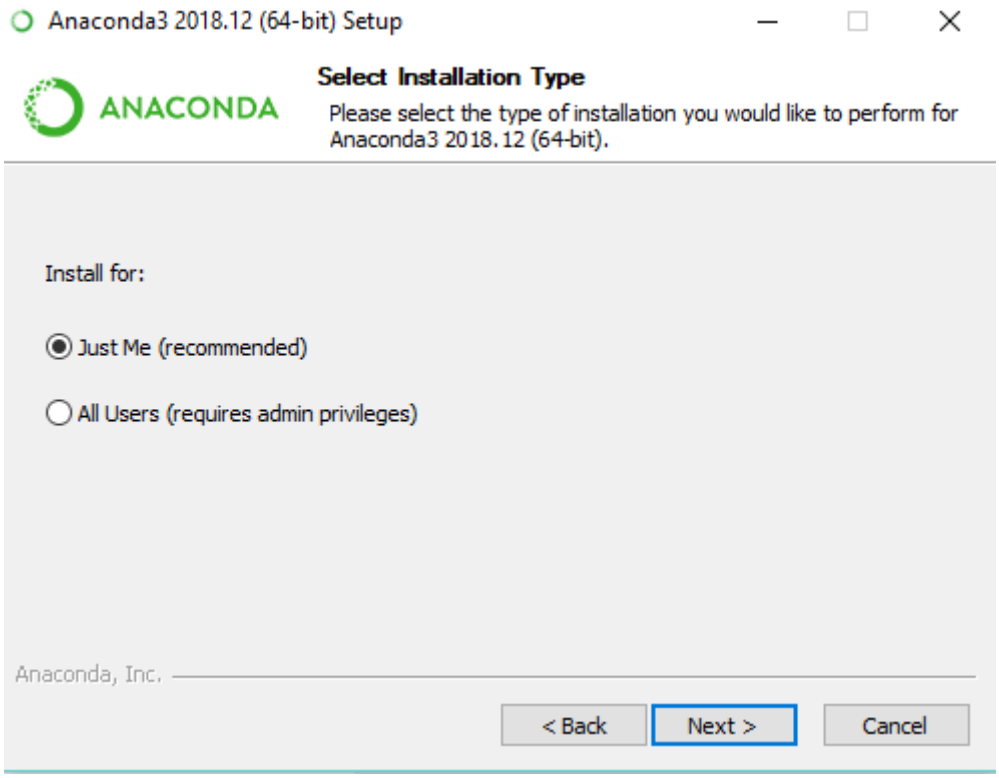
Una vez descargado, abrir la aplicación Anaconda.exe

Aparecerá la siguiente pantalla de bienvenida, clic en “Next”

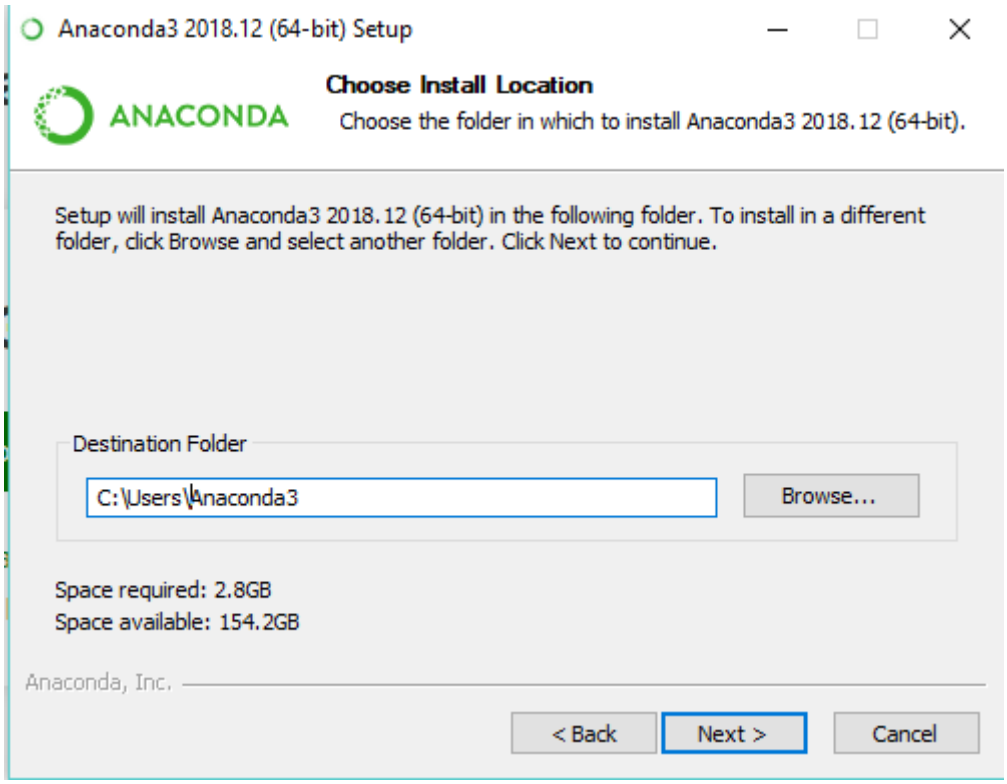
Luego aparecerá la pantalla de Términos y Condiciones, clic en “I Agree”



Luego aparecerá la pantalla de tipo de instalación, clic en “Next”

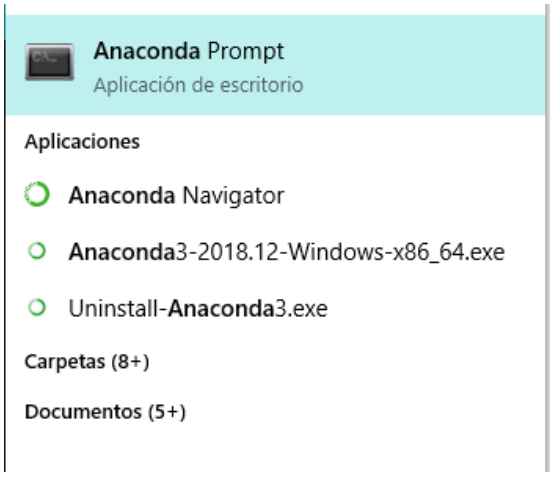


Luego aparecerá la pantalla para escoger la ruta de guardado de Anaconda, elegir la ruta y clic en “Next”



Anaconda se instalará en la ruta seleccionada. Una vez instalada se podrá hacer uso de su consola.

Para abrir la consola de Anaconda buscar aplicación “Anaconda Prompt” y dar clic.



Apéndice B. Manual de Usuario

Objetivo

Ayudar al usuario a realizar, mediante una guía, el proceso del flujo de trabajo en la herramienta Jupyter Notebook.

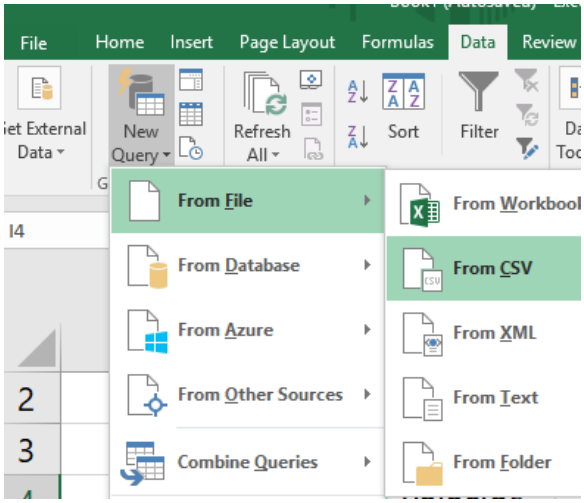
Descargar base de datos de proyectos Kickstarter

Navegar a página web https://webrobots.io/kickstarter-datasets/ en donde encontraremos todas las base de datos en formato .csv de los proyectos Kickstarter.

Modificar archivos csv en Excel

Para modificar archivos .csv en Excel, se debe seguir los siguientes pasos:

- Clic en Datos
- Nueva Consulta – Desde Archivo - CSV
- Elegir ruta de archivo csv
- Cargar



Una vez cargada la información, se mostrará como tabla en Excel. En cada hoja de podrá subir diferentes archivos csv y se los podrá agrupar.

Conversión de Formato UNIX a Fecha en archivo de data Excel.

Para cambiar el formato de fecha UNIX a Fecha utilizamos la siguiente fórmula:

“ =MULTIPLO.INFERIOR([@[created_at]]/60/60/24;1) +

FECHA(1970;1;1)”;

Siendo @[created_at] la fecha en formato de UNIX del tiempo en el cual se creó el proyecto Kickstarter.

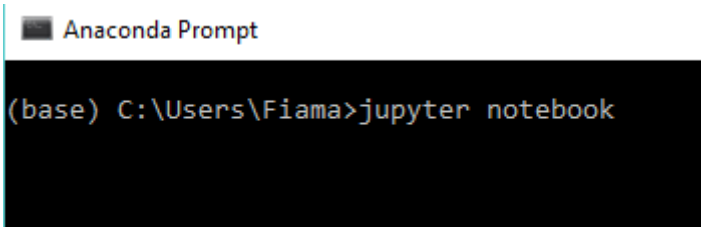
Exportar archivo Excel a CSV

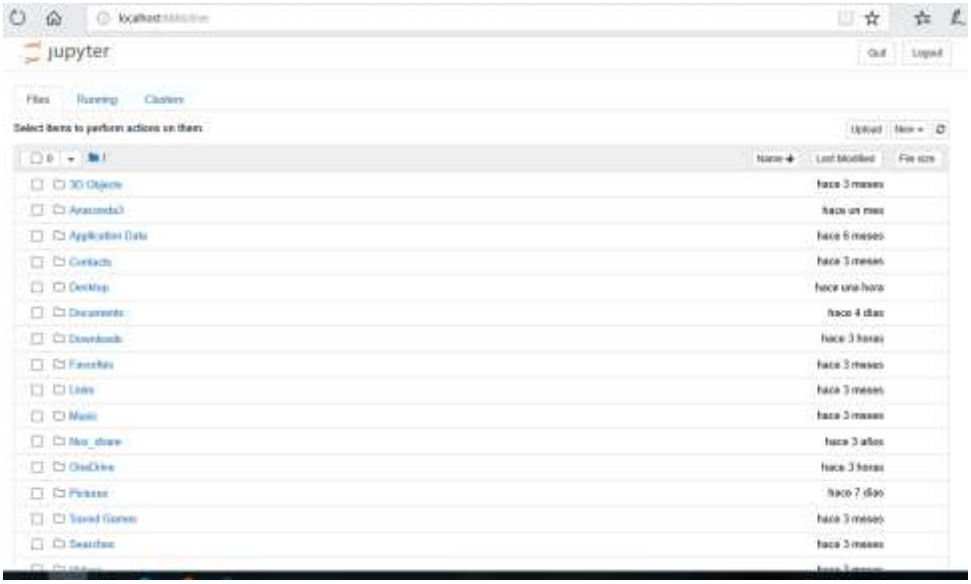
Para exportar el archivo Excel a CSV se debe usar el plugin XLTools para Excel.

Abrir Jupyter Notebook

Para abrir Jupyter Notebook, se necesita tener instalado la distribución Anaconda (Apéndice B).

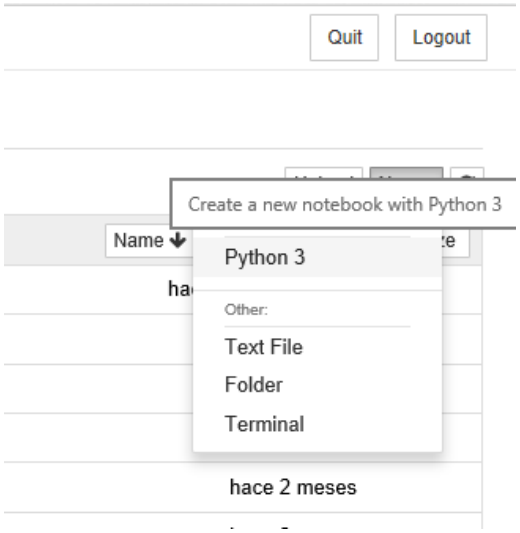
Abrir consola de Anaconda llamada “Anaconda Prompt” y escribir el comando “jupyter notebook”. Este comando abrirá la herramienta Jupyter Notebook en el navegador predeterminado. Nota: No cerrar la consola.





Crear Notebook

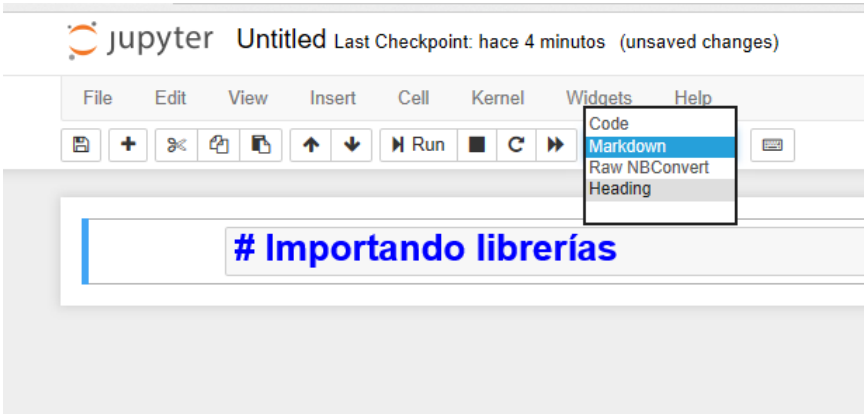
Para crear el notebook, el cual será el área donde realizaremos nuestro trabajo, seleccionar la ruta en donde se quiere trabajar y dar clic en el botón “New” y seleccionar “Python 3”



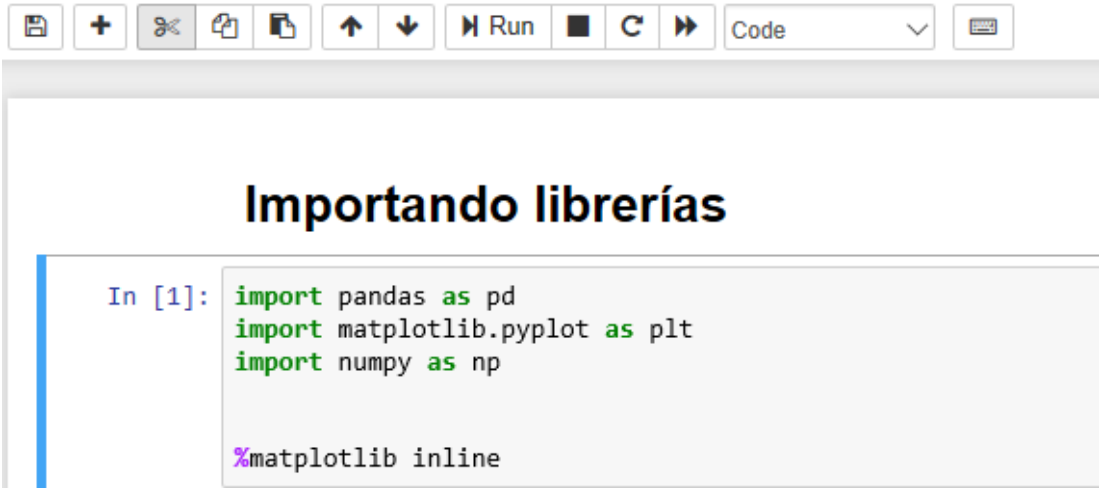
Textos y códigos

En el área de trabajo manejaremos dos tipos de input: texto y código Python.

Para ingresar texto elegir la opción Heading e ingresar el texto. El tamaño se puede cambiar de acuerdo al número de “#” que se ingrese delante del texto. Una vez ingresado aplastar la combinación de botones Shift + Enter.



Para ingresar código Python elegir la opción Code e ingresar el código. Una vez ingresado el código aplastar la combinación de botones Shift + Enter para procesar el código y presentar el resultado si se requiere.



Interpretar todas las celdas escritas

Para que la herramienta interprete el texto o código y poder mostrar los resultados de todos los campos ingresados, dar clic en el botón  y elegir la opción

Importar Librerías

Para importar las librerías necesarias para el proceso del diseño del modelo, ingresar el siguiente código:

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

%matplotlib inline

Cargar data

Para importar el archivo CSV con los datos de los proyectos Kickstarter a la herramienta Jupyter Notebook, ingresar el siguiente código:

dataframe = pd.read_csv("./kickstarter-demo2.csv")

Siendo lo que está dentro del paréntesis la ruta en donde se encuentra el archivo .csv. Para verificar que se cargó la data correcta se puede insertar el siguiente código:

dataframe.head(5)

Se podrá visualizar los cinco primeros proyectos de la base en una tabla.

Eliminar variables

Para eliminar variables que no son parte del grupo escogido usamos el siguiente código:

del dataframe['first_project']

Siendo el nombre de la variable el texto dentro de las llaves y comillas simples [‘ ‘] .

Verificar correlación de datos

Para verificar la correlación de datos, ingresar el siguiente código:

plot_corr(dataframe)

Se generará una imagen donde se compararán todas las variables unas a otras, cuando las variables se comparan a sí mismas habrá correlación por defecto por lo que se pueden ignorar. Normalmente el cuadro se pintará de color amarillo si existe correlación.

Si existen variables con correlación, es decir existe un cuadro de color amarillo en la imagen generada, eliminar una de ellas mediante el paso “Eliminar variables” para eliminar la correlación.

Cambiar valores True or False

Para cambiar todos los valores True a 1 y los valores False a 0 de las variables “state”, “disable_communication” y “spotlight”, ingresar el siguiente código:

```
state_map = {True : 1, False : 0}

dataframe['state'] = dataframe['state'].map(state_map)

dataframe['disable_communication'] =
dataframe['disable_communication'].map(state_map)

dataframe['spotlight'] = dataframe['spotlight'].map(state_map)
```

Validar porcentajes de proyectos exitosos y proyectos fallidos

Para validar el balance de porcentajes entre proyectos exitosos y proyectos fallidos en nuestro archivo cargado, ingresar el siguiente código:

```
num_true = len(dataframe.loc[dataframe['state']== True])

num_false = len(dataframe.loc[dataframe['state']== False])

print("Proyectos exitosos: {0} ({1:2.2f}%)".format(num_true,
(num_true/ (num_true + num_false))*100))

print("Proyectos fallidos: {0} ({1:2.2f}%)".format(num_false,
(num_false/ (num_true + num_false))*100))
```

Una vez ingresado se presentará el porcentaje y el número de proyectos exitosos y fallidos. En caso de que haya una diferencia muy alta entre los proyectos exitosos y fallidos, modificar la base de datos.

Separar datos en datos de entrenamiento y datos de evaluación.

Para separar el 30% de los datos a evaluación ingresar el siguiente código:

```
from sklearn.cross_validation import train_test_split

feature_col_names = ['disable_communication', 'goal', 'spotlight', 'duration',
'product_video', 'marketing_video', 'number_of_promises',
```

```
'frequented_questions', 'updates', 'product_explanation',
'about_us_explanation', 'promoted', 'picture_gallery',
'maximum_contribution', 'minimum_contribution', 'limited_courier']

predicted_class_names = ['state']

X = dataframe[feature_col_names].values

y = dataframe[predicted_class_names].values

split_test_size = 0.30

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=split_test_size,
random_state=42)
```

Siendo las variables seleccionadas el arreglo feature_col_names y split_test_size el porcentaje en decimal asignado a evaluación.

Validar porcentajes de datos exitosos o fallidos en entrenamiento y evaluación.

Para validar los porcentajes de proyectos exitosos y fallidos luego de haber separado la data en entrenamiento y evaluación, ingresar el siguiente código:

```
print ("{0:0.2f}% en entrenamiento
set".format((len(X_train)/len(dataframe.index))*100))

print ("{0:0.2f}% en evaluación

set".format((len(X_test)/len(dataframe.index))*100))

print ("")

print ("Entrenamiento exitosos :  {0}
({1:0.2f}%)".format(len(y_train[y_train[:] == 1]), (len(y_train[y_train[:]
== 1])/len(y_train))*100))

print ("Entrenamiento fallidos : {0}
({1:0.2f}%)".format(len(y_train[y_train[:] == 0]), (len(y_train[y_train[:]
== 0])/len(y_train))*100))

print ("")
```

```
print ("Evaluación Exitosos:      {0}
({1:0.2f}%)".format(len(y_test[y_test[:] == 1]), (len(y_test[y_test[:] ==
1])/len(y_test))*100))

print ("Evaluación Fallidos :      {0}
({1:0.2f}%)".format(len(y_test[y_test[:] == 0]), (len(y_test[y_test[:] ==
0])/len(y_test))*100))
```

Entrenar al modelo con data de entrenamiento mediante algoritmo Naive Bayes

Para alimentar al modelo de predicción mediante el algoritmo Naive Bayes, ingresar el siguiente código:

```
from sklearn.naive_bayes import GaussianNB

#create Gaussian Naive Bayes model object and train it with the data

nb_model= GaussianNB()

nb_model.fit(X_train, y_train.ravel())

nb_predict_train = nb_model.predict(X_train)

from sklearn import metrics

#training metrics

print("Exactitud: {0:.4f}".format(metrics.accuracy_score(y_train,
nb_predict_train)))
```

Se presentará el porcentaje en decimales de la exactitud del modelo alimentado con data de entrenamiento mediante el algoritmo Naive Bayes.

Verificar predicción del modelo con data de evaluación mediante algoritmo Naive Bayes

Para evaluar al modelo de predicción mediante la data de evaluación, ingresar el siguiente código:

```
# predict values using the testing data

nb_predict_test = nb_model.predict(X_test)

from sklearn import metrics

#training metrics
```

```
print("Accuracy: {0:.4f}".format(metrics.accuracy_score(y_test,
nb_predict_test)))
```

Se presentará el porcentaje en decimales de la exactitud del modelo alimentado con data de entrenamiento mediante el algoritmo Naive Bayes.

Entrenar al modelo con data de entrenamiento mediante algoritmo Naive Bayes

Para alimentar al modelo de predicción mediante el algoritmo Naive Bayes, ingresar el siguiente código:

```
from sklearn.ensemble import RandomForestClassifier

rf_model = RandomForestClassifier(random_state=0)

rf_model.fit(X_train, y_train.ravel())

rf_predict_train = rf_model.predict(X_train)

# training metrics

print("Exactitud: {0:.4f}".format(metrics.accuracy_score(y_train,
rf_predict_train)))
```

Se presentará el porcentaje en decimales de la exactitud del modelo alimentado con data de entrenamiento mediante el algoritmo Random Forest

Verificar predicción del modelo con data de evaluación mediante algoritmo Random Forest

Para evaluar al modelo de predicción mediante la data de evaluación, ingresar el siguiente código:

```
rf_predict_test= rf_model.predict(X_test)

print("Accuracy: {0:.4f}".format(metrics.accuracy_score(y_test,
rf_predict_test)))
```

Se presentará el porcentaje en decimales de la exactitud del modelo alimentado con data de entrenamiento mediante el algoritmo Random Forest.

Crear Ranking de Variables más importantes para el algoritmo Random Forest

Para crear un ranking de variables de mayor peso a la hora del éxito de un proyecto generado por el modelo entrenado con el algoritmo Random Forest, ingresar el siguiente código:

```
estimator = rf_model.estimators_[6]
from sklearn.tree import export_graphviz
export_graphviz(estimator, out_file='tree.dot',
        feature_names = feature_col_names,
        rounded = True, proportion = False,
        precision = 1, filled = True)
important_features = RandomForestClassifier()
important_features = important_features.fit(X_train,y_train)
importances = important_features.feature_importances_
std = np.std([tree.feature_importances_ for tree in
important_features.estimators_], axis=0)
ind = np.argsort(importances)[::-1]
print("Ranking:")
for f in range(X_train.shape[1]):
        print("%d. variable %d (%f)" % (f +1, ind[f],
importances[ind[f]]))
```

Se presentará una lista de variables siendo el mejor ranking el número 1. Las variables se identificarán con un número el cual es determinado por la posición en que se encuentran en la base de datos.

Para visualizar el orden de las variables ingresar el código: **dataframe.head(5)**

La primera variable empezará en la posición 0.

Dibujar el diagrama de árbol de decisiones del algoritmo Random Forest

Para visualizar el diagrama de árbol de decisiones, ingresar el siguiente código:

```
import graphviz
```

```
from IPython.display import display

with open("tree.dot") as f:

    dot_graph = f.read()

    display(graphviz.Source(dot_graph))
```

El diagrama se generará dentro de la herramienta en formato svg.

Entrenar al modelo de predicción con las variables más importantes usando el algoritmo de Random Forest

Para entrenar al modelo de predicción de éxito mediante el algoritmo de Random Forest y solo usando las variables más importantes, ingresar el siguiente código:

```
from sklearn.cross_validation import train_test_split

feature_col_names = ['number_of_promises', 'minimum_contribution', 'updates', 'goal', 'duration']

predicted_class_names = ['state']

X = dataframe[feature_col_names].values

y = dataframe[predicted_class_names].values

split_test_size = 0.01

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=split_test_size, random_state=42)from sklearn.ensemble import RandomForestClassifier

rf_model2 = RandomForestClassifier(random_state=0)

rf_model2.fit(X_train, y_train.ravel())

rf_predict_train2 = rf_model2.predict(X_train)

# training metrics

print("Exactitud: {0:.4f}".format(metrics.accuracy_score(y_train, rf_predict_train2)))
```

Prediciendo proyectos de financiamiento en masa mediante el modelo de predicción de éxito ingresando como input las variables más importantes.

Para poder conocer si un proyecto tendrá éxito o no, ingresando como input las variables más importantes, ingresar el siguiente código:

```
#numero_de_promesas, contribucion minima, actualizaciones, meta, duracion
number_of_promises='4'
minimum_contribution='30'
updates='1'
goal='200000'
duration='90'
input=[number_of_promises,minimum_contribution,updates,goal,duration]
input = np.array(input).reshape((1,-1))
outcome=rf_model2.predict(input)
print('1 = exito, 0 = fallido')
print(outcome)
#output:
```

Se presentará el número 1 si el proyecto tiende a tener éxito o el número 0 si el proyecto tiende a no tener éxito. Para ingresar valores se debe modificar el valor que se encuentra después de la variables después del signo igual y entre comillas simples number_of_promises, mínimum_contribution:updates, goal y duration.

Apéndice C. Manual Técnico

Objetivo

Ayudar al usuario a identificar los campos de la base de datos de proyectos tecnológicos en Kickstarter y sus valores correspondientes.

Base de datos

La base de datos está constituida por las variables que se escogieron después del análisis de los proyectos tecnológicos en Kickstarter. El archivo de la base de datos está en formato .csv, el cual puede ser consultado mediante Microsoft Excel. A continuación, se detallas los campos y sus contenidos:

Nombre de campo	*Contenido*
name	Nombre del Proyecto en Kickstarter.
disable_communication	Indica si se deshabilito comunicación con el dueño del proyecto. El campo será 1 si es verdadero y 0 si es falso.
spotlight	Indica si se usó las herramientas web de Kickstarter para promocionar el proyecto. El campo será 1 si es verdadero y 0 si es falso.
staff_pick	Indica si el proyecto fue elegido por Kickstarter como destacado (este campo es luego eliminado por correlación con el estado del proyecto). El campo será 1 si es verdadero y 0 si es falso.
duration	Indica la duración del financiamiento masivo para el proyecto. El campo será el numero en días.
product_video	Indica si el proyecto contiene un video instructivo del producto o servicio. El campo será 1 si es verdadero y 0 si es falso.
marketing_video	Indica si el proyecto contiene un video publicitario tipo propaganda para el producto o servicio. El campo será 1 si es verdadero y 0 si es falso.
number_of_promises	Indica el número de propuestas del proyecto para que los usuarios puedan contribuir con la cantidad de dinero

Nombre de campo	*Contenido*
	especificada en la propuesta. El campo ser la cantidad en números.
frequented_questions	Indica si el proyecto tiene una sección de preguntas frecuentes para los usuarios interesados. El campo será 1 si es verdadero y 0 si es falso.
updates	Indica si el proyecto tiene una sección de actualizaciones de decisiones y planes del proyecto para interés de los usuarios. El campo será 1 si es verdadero y 0 si es falso.
product_explanation	Indica si el proyecto tiene una sección explicando la idea y el beneficio del proyecto. El campo será 1 si es verdadero y 0 si es falso.
about_us_explanation	Indica si el proyecto tiene una sección explicando los antecedentes y quien es el dueño del proyecto. El campo será 1 si es verdadero y 0 si es falso.
promoted	Indica si el proyecto está siendo impulsado por compañías terceras. El campo será 1 si es verdadero y 0 si es falso.
mínimum_contribution	Indica el valor del aporte mínimo que un usuario puede dar al proyecto. El campo será el valor en números.
maximum_contribution	Indica el valor del aporte maximo que un usuario puede dar al proyecto. El campo será el valor en números.
picture_gallery	Indica si el proyecto cuenta con una galería de imágenes del producto o servicio. El campo será 1 si es verdadero y 0 si es falso.
limited_courier	Indica si el proyecto tiene limitaciones geográficas a la hora de entregar las promesas. El campo será 1 si es verdadero y 0 si es falso.
state	Indica si el proyecto fue exitoso o fallido.

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Yagual López Luis Manuel** con C.C.: # 0925470411, autor del trabajo de titulación: **Diseño de un Modelo de Predicción de Éxito para Proyectos Tecnológicos con Financiación en Masa Aplicando Técnicas de *Machine Learning,*** previo a la obtención del título de **INGENIERO EN SISTEMAS COMPUTACIONALES** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de graduación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 11 de marzo del 2019

Yagual López Luis Manuel
C.C: 0925470411

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA			
FICHA DE REGISTRO DE TESIS/TRABAJO DE GRADUACIÓN			
TÍTULO Y SUBTÍTULO:	Diseño de un Modelo de Predicción de Éxito para Proyectos Tecnológicos con Financiación en Masa Aplicando Técnicas de *Machine Learning.*		
AUTOR:	Yagual López Luis Manuel		
TUTOR:	Ing. Vicente Gallardo Posligua, Mgs.		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Ingeniería		
CARRERA	Ingeniería en Sistemas Computacionales		
TÍTULO OBTENIDO:	Ingeniero en Sistemas Computacionales		
FECHA DE PUBLICACIÓN:	11 de marzo del 2019	**No. DE PÁGINAS:**	82
ÁREAS TEMÁTICAS:	Software, Sistemas, Aprendizaje automático, Inteligencia artificial		
PALABRAS CLAVE:	Aprendizaje automático, proyectos tecnológicos, Kickstarter, modelo de predicción, Python, Jupyter Notebook		

RESUMEN:
Los proyectos con financiación en masa se han vuelto una tendencia en el emprendimiento a nivel global mediante el uso de plataformas tales como Kickstarter, pero estos conllevan un alto riesgo de que no logren la meta a recaudar por lo que es importante el análisis y preparación del proyecto antes de su lanzamiento; por este motivo se propone el desarrollo de un modelo de predicción de éxito para proyectos tecnológicos con financiación en masa para ayudar en la toma de decisiones previo al lanzamiento y publicación del proyecto en la plataforma Kickstarter. Para el proyecto se utilizó la investigación cualitativa, descriptiva con análisis documental como técnica de recolección de datos. Se analizaron documentos relacionados a construcción de modelos predictivos con algoritmos de aprendizaje automático y se analizaron las variables que influyen en el éxito de los proyectos en la plataforma de Kickstarter y se consolidó información de los proyectos tecnológicos de la plataforma mediante bases externas e ingreso manual. Se diseñó un flujo de trabajo, basado en prácticas generales, para el diseño del modelo predictivo y se escogieron los algoritmos de aprendizaje automático enfocados al resultado a obtener. Una vez diseñado el modelo de predicción con los algoritmos escogidos, se evaluó la exactitud y se comprobó la confiabilidad de al menos un 70% en las predicciones de éxito para los proyectos tecnológicos con financiación en masa.

ADJUNTO PDF:	☒ SI	☐ NO
CONTACTO CON AUTOR:	**Teléfono:**+593-4-2175148 / 0959240093	E-mail: lucho15465@gmail.com
CONTACTO CON LA INSTITUCIÓN:	**Nombre:** Ing. Edison José Toala Quimí	
	Teléfono: +593-042 20 27 63 / 593-9-90976776	
	E-mail: edison.toala@cu.ucsg.edu.ec	

SECCIÓN PARA USO DE BIBLIOTECA	
Nº. DE REGISTRO (en base a datos):	
Nº. DE CLASIFICACIÓN:	
DIRECCIÓN URL (tesis en la web):	