UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE CIENCIAS

TESIS

PYMACH Y SPARKMACH: SISTEMAS DE PROCESAMIENTO DE DATOS CON DIMENSIÓN VARIABLE USANDO ALGORITMOS DE MACHINE LEARNING

PARA OBTENER EL TÍTULO PROFESIONAL DE:

LICENCIADO EN CIENCIA DE LA COMPUTACIÓN

ELABORADA POR:

GUSSEPPE JESÚS BRAVO ROCCA

ASESOR:

Dr. JOSÉ ANTONIO FIESTAS IQUIRA

LIMA-PERÚ

2019

"Si no puedo dibujarlo, es que no lo entiendo."

Albert Einstein

Dedicado a mi mamá, mi hermana y la ciencia...

Agradecimientos

Los agradecimientos recaen en los profesores que dieron parte de su tiempo para compartir sus conocimientos conmigo y el apoyo en cada paso de este proyecto...

Resumen

La presente tesis propone dos sistemas de análisis y predicción de datos enfocados a problemas relacionados al Machine Learning : *Pymach* y *Sparkmach*. Este sistema conjunto tiene el fin de reducir y automatizar los pasos convencionales que conlleva la creación de un modelo predictivo en general. Para este fin se hace uso de técnicas de inteligencia artificial, particularmente, Machine Learning, para crear modelos a medida que puedan predecir eventos a futuro, en aplicaciones tales como, lugares y frecuencias de accidentes de tránsito, localización, tiempos de espera de autobuses, consumo de combustible, entre otros. Para ello, se ha trabajado con datos simulados y reales que, junto al sistema, se han desplegado en un clúster de CPUs. Debido a la ingente cantidad de datos, se ha trabajado con técnicas de paralelismo y Big Data para el procesamiento eficiente de los mismos. Finalmente, *Pymach* y *Sparkmach*, escrito en Python y PySpark respectivamente, están desplegadas en una aplicación web para la interacción con el usuario.

Índice

Índice de figuras

Índice de tablas

Abreviaciones

ML	**M**achine **L**earning
IA	**I**nteligencia **A**rtificial
CPU	**C**entral **P**rocessing **U**nit
HDFS	**H**adoop **D**istributed **F**ile **S**ystem
CSV	**C**omma **S**eparated **V**alue
TXT	**T**ext File
MIT	**M**assachusetts **I**nstitute of **T**echnology
KB	**K**ilo **B**yte
PCA	**P**rincipal **C**omponent **A**nalysis
SVC	**S**upport **V**ector **M**achine
LDA	**L**inear **D**iscriminant **A**nalysis
YARN	**Y**et **A**nother **R**esource **N**egotiator
RAM	**R**andom **A**ccess **M**emory

Capítulo 1

Introducción

El advenimiento de una gran variedad de datos que provienen de diferentes fuentes, como sensores, dispositivos móviles, búsquedas en línea, redes sociales, bolsa de valores, tendencias del mercado, etc., ha conllevado a una demanda de profesionales y herramientas que puedan extraer, procesar, analizar y dar valor a esos datos que un principio no brindaban un conocimiento adicional que su misma información en sí, es decir, no se aprovechaba los patrones que podían surgir al ver la data como un todo y no individualmente. Actualmente para poder extraer nueva información y patrones de una data en general se hace uso de técnicas estadísticas y de Machine Learning para describir la data y crear modelos a partir de estos. Todo esto lo realiza un equipo de científicos de datos que van creando preguntas a partir del análisis de los datos, y junto con las personas que saben el significado de los datos (el rubro o negocio), crean un producto (modelo) que ayuda a mejorar la toma de decisiones [18].

1.1. Motivación

En los últimos 10 años la revolución digital ha dado lugar a herramientas económicas y accesibles de recolección y almacenamiento de datos. Este crecimiento desmesurado de datos ha provocado que el proceso de extracción de conocimiento sea más difícil, y en su defecto, provoque problemas de escalabilidad y rendimiento.

El proceso de extraer, procesar, analizar y predecir nuevos valores a partir de estos datos, es un conjunto de pasos sistemáticos que requiere un conocimiento del campo del problema (por ejemplo, tráfico vehicular), requiere técnicas de estadística, técnicas de Machine Learning, entre otras. Dado ello, se abre la posibilidad de poder automatizar ciertos pasos generales y dejar los pasos que requieran un trabajo más minucioso al usuario, logrando una disminución considerable en los tiempos de desarrollo.

Por otro lado, la toma de decisiones no solo se realiza sobre los datos que provienen de un banco histórico de meses o años, sino también de datos en tiempo real o a una velocidad variable. Para dicho escenario, el sistema predictivo no sólo tiene que ser capaz de modelar en un tiempo reducido sino también actuar con grandes cantidades de datos. Por ejemplo, la creciente demanda de modelos predictivos para dar sentido a los valores medidos por sensores en una ciudad inteligente requiere de nuevos métodos ágiles de desarrollo de sistemas de predicción que estén preparados para actuar ante cualquier cambio de estructura, velocidad y volumen de datos, para tal efecto, disponer de una tecnología que pueda automatizar sistemáticamente el proceso de modelar un problema es vital para reducir los tiempos de procesamiento en un centro de datos.

La automatización de un modelo de Machine Learning no tiene que ser estática, es decir, tiene que ser lo suficientemente dinámica para amoldarse a cualquier tipo de datos, para ello debe aprender de estructuras de problemas pasados, lo que se llamaría meta-aprendizaje. Para ello el mayor esfuerzo de tal sistema debe enfocarse en poder almacenar la estructura raíz de los datos (variables numéricas, categóricas, fechas, etc.) y tener claro cómo lidiar con ellos para maximizar la predictibilidad de un modelo.

Adicionalmente, es interesante destacar que mucho del trabajo realizado en esta tesis proviene de trabajados publicados en revistas y conferencias sobre diferentes aplicaciones de ML en localización de interiores, Big Data [13], y seguridad ciudadana [19].

1.1.1. Contribuciones científicas

- Revista. Impact factor: 2.057, Q2. "Manuel Castillo-Cara, Jesús Lovón-Melgarejo, Gusseppe Bravo-Rocca et al. **An Analysis of Multiple Criteria and Setups for Bluetooth Smartphone-Based Indoor Localization Mechanism**. Journal of Sensors, vol. 2017, Article ID 1928578, 22 pages

- Revista.Impact factor: 2.677, Q1. "Manuel Castillo-Cara, Jesús Lovón-Melgarejo, Gusseppe Bravo-Rocca et al. **An Empirical Study of the Transmission Power Setting for Bluetooth-Based Indoor Localization Mechanisms**. Sensors. 17(6):1318.

- Conferencia. SCOPUS. Gusseppe Bravo-Rocca, Piero Torres-Robatty et al. **Sparkmach: A Distributed Data Processing System Based on Automated Machine Learning For Big Data**. Springer Communications in Computer and Information Science (CCIS) Series.

- Conferencia. SCOPUS. Jesús Lovón-Melgarejo, Manuel Castillo-Cara, Gusseppe Bravo-Rocca et al. **Supervised learning algorithms for indoor localization**

fngerprinting using BLE4.0 Beacons. 4th IEEE Latin American Conference on Computational Intelligence LA-CCI. Arequipa, Perú.

- Conferencia. SCOPUS. "**Gusseppe Bravo Rocca, Manuel Castillo-Cara et al. Citizen security using Machine Learning algorithms through Open Data**. 8th IEEE Latin- American Conference on Communications LATINCOM. Medellin, Colombia.

1.2. Objetivos

El objetivo general es desarrollar un paquete de código libre (open-source) que ayude a acelerar la creación de un modelo basado en Machine Learning. Dicho paquete solo necesitará los datos del usuario y la variable a predecir. A partir de este contexto, el programa trabajará de manera independiente mostrando los resultados de una forma autoguiada.

Los objetivos particulares son los siguientes:

- Estudiar los métodos de análisis de datos y algoritmos de Machine Learning.
- Implementar un módulo del análisis exploratorio de los datos.
- Implementar un módulo de preprocesamiento y selección de características.
- Desarrollar un módulo de evaluación y optimización de modelos mediante la búsqueda de los mejores hiperparámetros.

1.3. Resumen de los capítulos

Con el fin de desarrollar los objetivos propuestos, el presente trabajo está constituido de 7 capítulos.

En el Capítulo 1, Introducción. Se presenta una breve introducción y motivación del problema, así como también, los objetivos a desarrollar.

En el Capítulo 2, Antecedentes y Estado del arte. Se introduce conceptos relacionados al Machine Learning, Big Data, Ciencia de datos y software de automatización.

En el Capítulo 3, Tecnologías subyacentes. Se hace un recuento de las tecnologías utilizadas en la implementación de Pymach y Sparkmach.

En el Capítulo 4, Pymach. Se da una introducción a la plataforma, sus funciones, su utilidad y su arquitectura.

En el Capítulo 5, Sparkmach. Presentación de la librería, motivación y diferencias con Pymach.

En el Capítulo 6, Experimentos y resultados. Presentación del uso de Pymach y Sparkmach en diferentes conjuntos de datos.

En el Capítulo 7, Conclusiones y trabajo futuro. Se indica las lecciones aprendidas y las posibles mejoras al presente trabajo.

Capítulo 2

Antecedentes y Estado del arte

En este apartado se brinda una introducción al campo de la Inteligencia Artificial, Machine Learning y conceptos relacionados, tales como, Big Data, Ciencia de datos y software de automatización. Asimismo, se realiza una descripción al campo al que pertenece el presente trabajo.

2.1. Inteligencia Artificial

La definición de Inteligencia Artificial (IA) no es única, sin embargo, se puede llegar a un concepto universal: IA intenta lograr que las computadoras realicen cosas en las cuales, hasta este momento, los humanos son mejores [20]. Para tal fin es necesario realizar tareas que solo un humano puede ser capaz de lograr.

El test de Turing propuesto por Alan Turing en 1950, indica que para que una máquina sea considerada inteligente, un humano no pueda diferenciar, entre unas respuestas escritas dado un conjunto de preguntas, si estas provienen de un humano o una máquina. Por lo tanto, para que una máquina pueda lograr dicho objetivo debe ser capaz de tener las siguientes habilidades:

- Procesamiento de lenguaje natural: Ser capaz de comunicarse en algun lenguaje.
- Representación del conocimiento: Almacenar lo que percibe.
- Razonamiento automatizado: Usar información almacenada para responder a preguntas e inferir conclusiones.
- Machine Learning: Detectar y extrapolar patrones para la adaptación con su entorno.

Un test más rigoroso o la versión más completa del test de Turing indica que si existiese una interacción humano-computador, entonces la máquina debe tener estas habilidades adicionales:

- Visión por computadora: Percibir objetos.
- Robótica: Capacidad de manipular y mover objetos.

Para ubicarnos un poco en el contexto de la automatización en el campo de la Inteligencia Artificial, se muestra el siguiente gráfico, figura 2.1. El presente trabajo se ubicaría dentro del campo *Machine Learning, Supervised Learning, Classic Models y Numeric Prediction.*

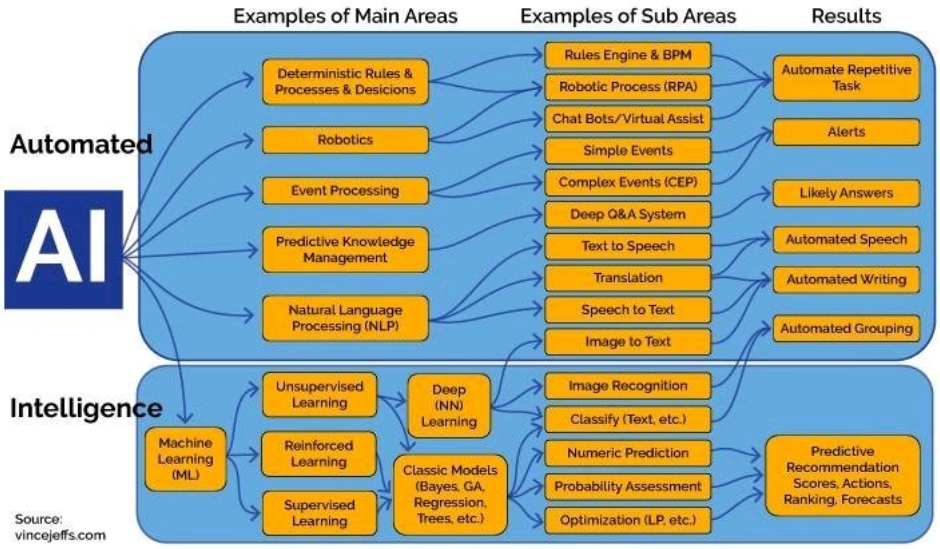


FIGURA 2.1: Tipos de automatización dentro de la Inteligencia Artificial [1].

2.1.1. Machine Learning

Una primera definición es dada por Tom M. Mitchell, que cita lo siguiente:

"Se dice que un programa de computadora aprende de la experiencia E, con respecto a una clase de tareas T y rendimiento P, si su rendimiento P en las tareas en T, mejora con la experiencia E". Fuente: Mitchel, 1997 [21].

Por ejemplo, si se quiere reconocer palabras escritas en una imagen se tendría lo siguiente:

- T (Task): Es la tarea para reconocer las palabras y clasificarlas dentro de una imagen.
- P (Perfomance): Mide cuántas palabras fueron correctamente clasificadas.
- E (Experience): Es todo el conjunto de datos donde aparecen las imagenes y su clasificación.

Por otro lado, una definición más larga sería la siguiente:

Básicamente el aprendizaje se divide en tres problemas: supervisado, no supervisado y por reforzamiento. En el primero se entiende como una función que mapea valores X a valores y; dicha función aprende de un conjunto de datos de entrenamiento del tipo $D = (X_i, y_i)_{i=1}^{N}$, donde D es una tupla que contiene las variables predictoras y la variable de respuesta y N es el número de filas.

La variable X puede ser una estructura compleja como una imagen, medidas de sensores, un mensaje de correo, etc. Sin embargo, y se asume que es una variable categórica (digamos, clases) tal que $y_i \in \{1, \ldots, C\}$ si se trata de un problema de clasificación, por otro lado si $y_i \in \mathbb{R}$, se trataría de un problema de regresión. En el segundo tipo, aprendizaje no supervisado, el objetivo es buscar patrones en la data sin la necesidad de tener la variable de respuesta y, donde solo se tiene $D = (X_i)_{i=1}^{N}$. Finalmente el tercer tipo, aprendizaje por reforzamiento, aprende a medida que se le de un premio o castigo por cada acto que un agente realice, estos problemas pertenecen particularmente al campo del Machine Learning profundo [22].

2.1.2. Ciencia de datos

No hay un acuerdo en tener una sola definición con respecto a ciencia de datos, pero se podría tomar como aquella área de trabajo que colecciona, prepara, analiza, visualiza, administra y preserva una larga cantidad de información [23]. La figura 2.2 muestra dicho flujo [2], en donde cada etapa es un proceso iterativo, es decir, se comienza con un prototipo inicial y luego éste se va mejorando en cada etapa .

Por otro lado, en la práctica se puede dividir en las siguiente etapas [24]:

- Entender la data.
- Preparar la data.

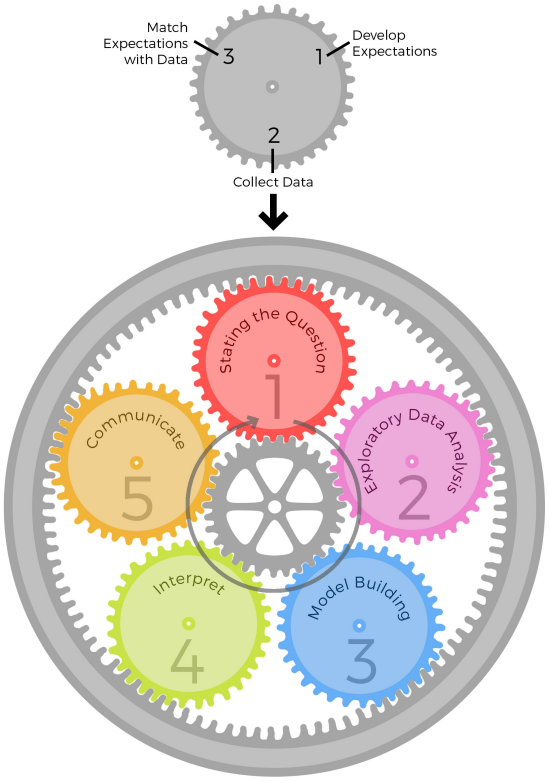


FIGURA 2.2: Ciclos del análisis en ciencia de datos. Fuente: Elizabeth M. Roger D. Peng [2].

- Selección de características.
- Modelamiento.
- Mejora de modelos.

Donde cada etapa de dicho flujo involucra un análisis especial dependiendo del problema que se quiera atacar y del campo de estudio.

La Ciencia de datos combina no solo Machine Learning, sino también, habilidades de programación, matemática y experiencia en el campo de estudio del problema a resolver. La figura 2.3 muestra las habilidades necesarias para dominarla.

2.1.3. Técnicas de automatización en algoritmos de Machine Learning

Tradicionalmente para atacar un problema de clasificación o regresión, el científico de datos tiene que pasar por una serie de etapas que van desde la limpieza de los datos hasta la mejora del modelo elegido. En los últimos años se ha visto un mayor incremento

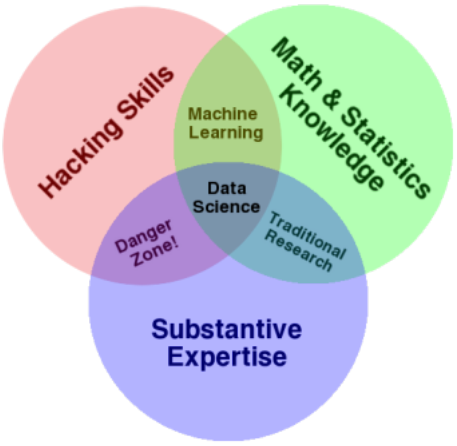


FIGURA 2.3: Ciencia de datos. Combinación de varias habilidades.

de herramientas que tratan de automatizar dicho proceso y que ha permitido al experto concentrarse en puntos más críticos del problema.

Gran parte de las ideas compartidas en los trabajos de aprendizaje automatizado involucran elegir qué algoritmo usar para un determinado problema, cómo preprocesar sus características y como buscar el mejor conjunto de hiperparámetros para dicho modelo [3].

Algunos trabajos realizados en este campo son los siguientes:

- Auto-WEKA, 2013

  Es el primer trabajo publicado e implementado en el campo de la automatización de selección de algoritmos e hiperparámetros, está basado en el software WEKA escrito en Java. Usa no más de 5 clasificadores en el proceso de 'ensemble' (optimización), también soporta selección de características.

- Hyperopt-sklearn, 2014

  Se desarrolló usando un enfoque similar a Auto-WEKA basado en la librería de ML: scikit-learn(2011). hyperopt-sklearn puede manejar solo pequeña y mediana data.

- MLbase, 2015

  Es el primer trabajo publicado e implementado en el campo de la automatización de selección de algoritmos e hiperparámetros que soporta computación distribuida en un clúster de computadores, está basado en Apache Spark - MLlib.

- AutoML, 2015

  Está basado en Auto-WEKA, pero combina una plataforma de Machine Learning paramétrico y optimización bayesiana. Ver figura 2.4.

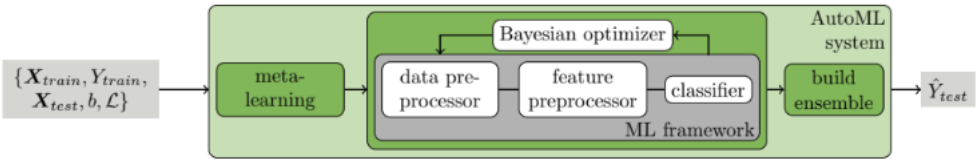


FIGURA 2.4: Flujo de automatización en AutoML [3].

Otros trabajos similares también son: TPOT, 2016; H2O AutoML, 2017; Google Cloud AutoML, 2018; Auto-Keras, 2018. Donde el objetivo común de todos ellos es obtener un conjunto de estimadores que puedan modelar un problema de Machine Learning de manera automática.

Capítulo 3

Tecnologías subyacentes

Para el desarrollo de los paquetes Pymach y Sparkmach se hizo uso de varias herramientas de código libre. Entre las cuales se encuentra el lenguaje de programación, Python, sus librerías y parte del software enfocado a Big Data.

3.1. Lenguaje de programación

3.1.1. Python

Python es un lenguaje de alto nivel diseñado para ser fácil de leer, debido a su sintaxis, y de fácil implementación. Es un lenguaje de código libre, lo que implica que es de uso libre tanto para propósitos personales y comerciales. Python es multiplataforma, el cual se ejecuta en sistemas operativos tales como Mac, Windows y Unix en general.[25]

Python es considerado un lenguaje interpretado y orientado a objetos con tipado dinámico lo que favorece a un rápido desarrollo de aplicaciones, asimismo, viene con muchas librerías incluidas para casi cualquier campo de estudio, tales como Biología, Física, Matemática, Inteligencia Artificial, Finanzas, y más; convirtiéndolo en un lenguaje multipropósito.

Según TIOBE [26], compañía que analiza el uso de los lenguajes de programación, Python ocupa el cuarto lugar como el lenguaje más usado a nivel mundial para el año 2017 y 2018 con un crecimiento anual consistente.

Dado que actualmente es el lenguaje más usado en el campo de la computación científica y el análisis de datos, los cuales siguen la misma línea que esta tesis, se ha optado por trabajar con este lenguaje.

3.2. Librerías

3.2.1. Numpy

NumPy es el paquete fundamental para la computación científica con Python. Esta librería se ha usado para el trabajo con matrices, en operaciones tales como, adición, multiplicación, transposición, uniones, etc. Adicionalmente, esta librería es requisito para el desarrollo de otras librerías que te abstraen de muchas líneas de código en una función de más alto nivel.

3.2.2. pandas

Librería que proporciona estructuras de datos de alto rendimiento y herramientas de análisis de datos. Se ha usado este paquete para el trabajo con *dataframes*, que es la estructura de datos predilecta en el mundo de ciencia de datos, tanto en Python, R o Spark. Además, contiene una colección de funciones para un análisis exploratorio de datos, así como también, gráficos estadísticos.

3.2.3. Scikit-learn

Scikit-learn es un módulo de Python que integra una gran variedad de algoritmos de Machine Learning tanto para problemas de pequeña y mediana escala supervisados y no supervisados, ver figura 3.1. Dado que éste se basa en el ecosistema científico de Python, es fácilmente compatible con las diferentes librerías que este lenguaje posee tales como numpy, scipy, pandas, etc.

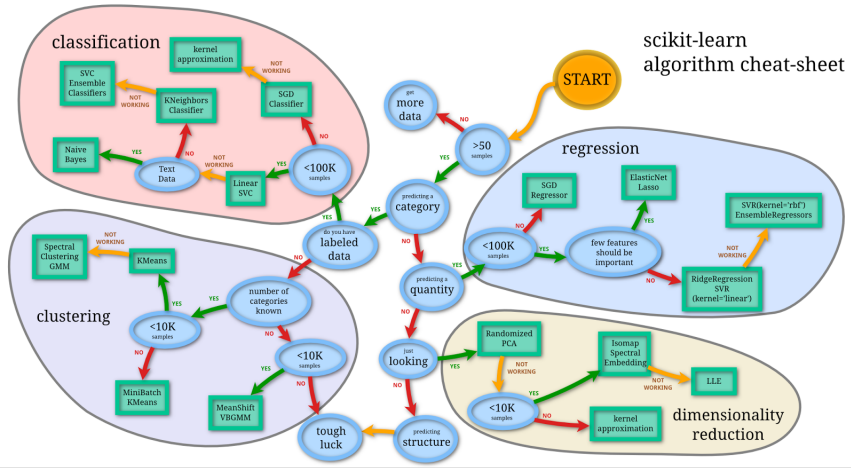


FIGURA 3.1: Algoritmos de Machine Learning en Scikit-learn [4].

Aunque la interfaz de scikit-learn está escrita en Python, ésta usa implementaciones en el lenguaje C para operaciones con matrices y arreglos, aumentando el rendimiento general.

Scikit-learn es ampliamente usado por grandes compañias tales como, INRIA, Spotify, Betaworks, Evernote, entre otras [27]. Convirtiéndola en la librería de Machine Learning más usada en la actualidad.

La orientación a objetos que posee esta librería permite crear aplicaciones altamente modulares, esto es, porciones de código con funciones específicas e independientes entre sí. Esta principal característica fue la escogida para implementar la presente librería.

3.2.4. Plotly

Librería que se encarga de la visualización de datos a pequeña, mediana y gran escala (con optimización) y en diferentes interfaces, tales como Python, R, Julia, etc. La diferencia de esta librería con las tradicionales como Matplotlib o Seaborn, está en que todos los gráficos generados por este paquete son dinámicos, ya que se basan en la tecnología d3.js de javascript la cual es la librería por defecto en cuestiones de gráficos de toda índole. Esto es muy útil ya que se puede mostrar datos adicionales simplemente pasando el mouse por cualquier punto del gráfico, asimismo, se hace uso de tecnologías *responsive*, haciendo que los gráficos puedan ser vistos en cualquier dispositivo, ya que estos se escalan de manera automática.

3.2.5. Flask

Es un *microframework* de desarrollo web escrito en python basado en Werkzeug y Jinja2. La elección de Flask en contraparte con otras plataforma de desarrollo se dió básicamente por la facilidad de uso y despliegue en pocos pasos. Con Flask fue posible desplegar todo el flujo de análisis de datos en una aplicación web, donde el usuario final podrá subir sus archivos y ejecutar cada módulo del presente trabajo a demanda.

3.3. Herramientas de procesamiento distribuido

3.3.1. Apache Spark

Spark es un motor de procesamiento de datos distribuido de propósito general basado en el paradigma *map reduce* de Hadoop. Una de sus mayores características es que las

ejecuciones de los trabajos lo hace en memoria, a diferencia de Hadoop, que lo hace en disco, reduciendo los tiempos hasta en un factor de 100. Spark cuenta con diferentes módulos, en particular se trabajó con su librería de aprendizaje automatico, MLlib y ML. Dichas librerías contienen una gran variedad de algoritmos para problemas de clasificación, regresión, agrupamiento (clustering) y más. Adicionalmente, Spark incluye también funciones para la limpieza y preprocesamiento de datos, todas estas funciones están pensadas con el concepto de escalabilidad, sea tanto en un computadora local con muchos núcleos o en un clúster de computadores. Ver figura 3.2. Spark se usó con el objetivo de escalar Pymach a grandes datos, usando su interfaz con Python, PySpark.

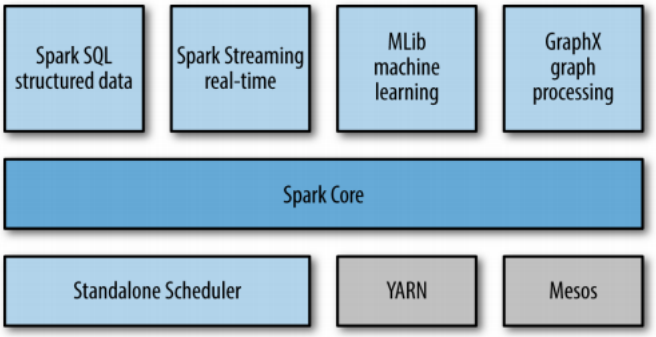


FIGURA 3.2: Plataforma Spark [5].

3.3.2. Apache Hadoop

Plataforma que permite el procesamiento distribuido de grandes cantidades de datos a través de clústeres de computadoras basado en disco. Hadoop cuenta con una gran variedad de módulos, en particular se ha trabajado con Hadoop HDFS, el cual aloja los datos de manera distribuida en todo el clúster y puede ser visto desde cualquier nodo, ayudando al paralelismo. Ver figura 3.3. El uso de Hadoop es importante debido a que brinda un ecosistema ideal para acoplar Apache Spark al mundo del Big Data.

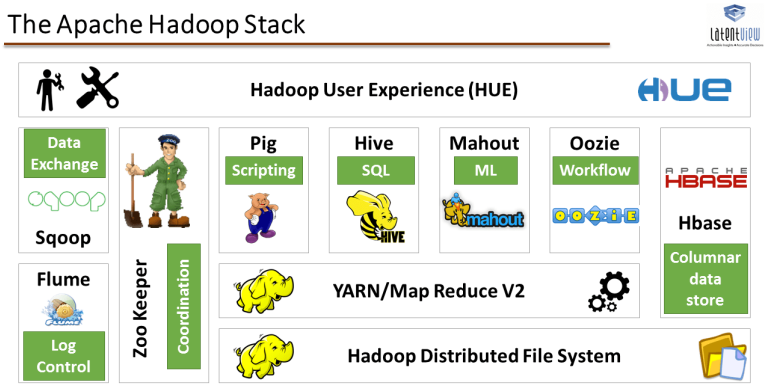


FIGURA 3.3: Plataforma Hadoop [6].

Capítulo 4

Pymach

En este apartado se habla sobre la estructura de Pymach y los módulos que la componen.

4.1. Definición

Pymach es una librería escrita en Python que analiza problemas de Machine Learning (PYthon MACHine learning) de una manera descriptiva, gráfica y resumida. El objetivo principal de Pymach es ofrecer un primer análisis automático para un conjunto de datos, esto es, dar un resumen general sobre la estructura (dimensionalidad, forma y tipado), patrones (correlaciones, agrupaciones y dispersamiento) y modelos (clasificación y regresión) de éstos. Logrando con ello, una primera impresión de los datos, para tener un enfoque general de los detalles que esta data esconde.

Algunas de las características más importantes de esta librería son las siguientes:

- No se necesita escribir ninguna línea de código, la ejecución de la librería es totalmente visual. La librería está desplegada en un servicio web.
- Solo necesita la data (con cierta limpieza), la variable que se quiere predecir y el tipo de problema (clasificacion o regresión), por ejemplo, un archivo en formato CSV y TXT.
- Es totalmente open source, con licencia MIT, lo que implica que puede ser copiado y modificado.
- La presentación de los resultados es interactiva, lo que facilita a la interpretación de los mismos. Se hace uso de una librería (plotly) que otorga los mejores gráficos adaptativos y configurables.

Pymach está dividido en 7 etapas, como se muestra en la figura 4.1, las cuales conforman un flujo de trabajo que va desde la entrada de datos hasta la presentación del dashboard de resultados. La elección de dicha arquitectura fue escogida con la finalidad de tener una librería altamente modular, lo que implica que al actualizar un módulo, éste pueda mejorar el rendimiento de todo el conjunto. Los pasos son los siguientes:

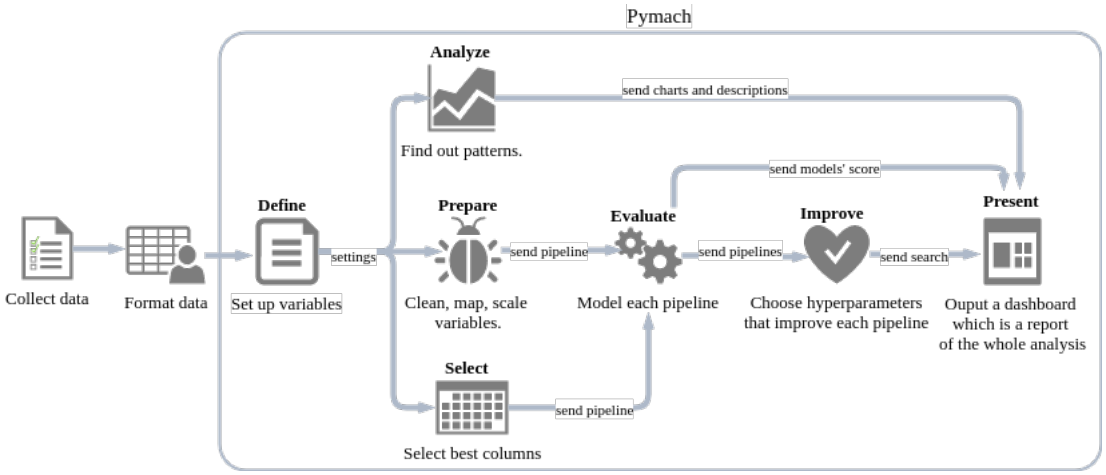


FIGURA 4.1: Flujo de trabajo de Pymach.

- **Define (D)**: Definir el problema.
- **Analyze (A)** : Se encarga del análisis exploratorio de la datos.
- **Prepare (P)** : Se encarga de la limpieza y escalamiento de los datos .
- **Select (S)** : Escoge las mejores variables predictoras.
- **Evaluate (E)** : Ejecuta los modelos.
- **Improve (I)** : Realiza una búsqueda de los mejores hiperparámetros.
- **Present (P)** : Presenta el dashboard de resultados.

Todos estos pasos pueden ser representados mediante un grafo de la siguiente manera:

Sea $G = (\hat{V}, \hat{E})$ un grafo dirigido o digrafo con un conjunto de vértices $\hat{V}$ y un conjuntos de aristas $\hat{E}$, de tal manera que $\hat{V} = \{D, A, P, S, E, I\}$, es decir, cada etapa es representado por un nodo. De la misma manera, el conjunto $\hat{E}$ está conformado por los pares ordenados $\{(D, A), (D, P), (D, S), (A, P), (P, E), (E, I), (I, P)\}$ como se muestra en la figura 4.2.

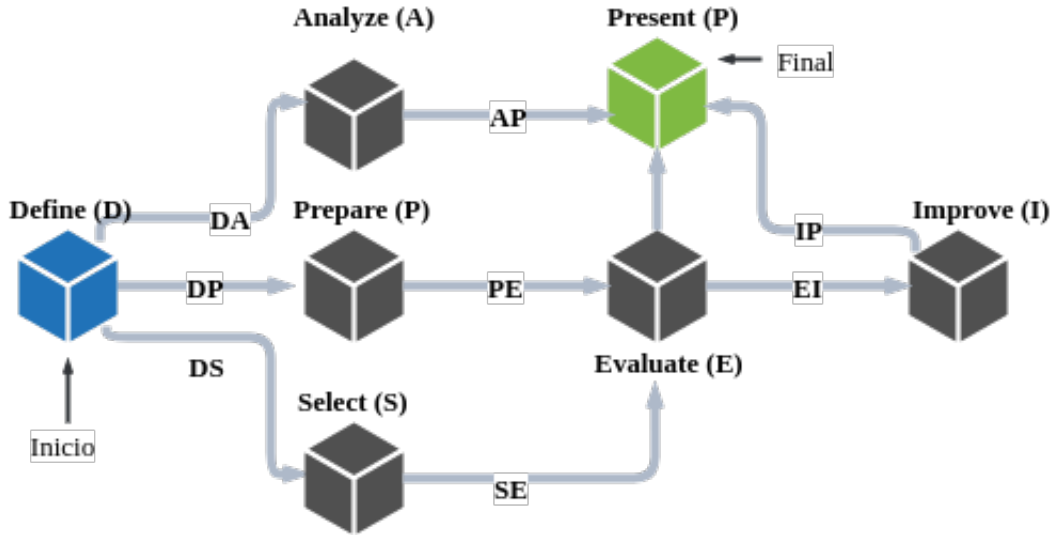


FIGURA 4.2: Representación de pymach mediante un digrafo

Esta estructura es muy útil para poder definir la relación que existe entre una etapa y otra, por ejemplo, la arista dirigida (D, A) tiene un significado distinto que la arista (P, E). Por otro lado, se puede extraer informacion útil de los caminos que se puedan encontrar, tales como (D, DP, P, PE, E) o (D, DA, A, AP, P). Adicionalmente cada nodo y arista tiene una significancia de por sí. Una definición más formal se muestra a continuación.

Definition 4.1. Sea $G = (\hat{V}, \hat{E})$ un grafo dirigido, que consiste en un conjunto $\hat{V}$ de nodos y un conjunto $\hat{E}$ de aristas. Donde $\hat{V} = \{D, A, P, S, E, I\}$ representa el conjunto de estados en un flujo de trabajo y $\hat{E} = \{(D, A), (D, P), (D, S), (A, P), (P, E), (E, I), (I, P)\}$ cada camino dentro de este flujo. Además, sea $\mathcal{L}$ una función de costo que minimice el error de un modelo de Machine Learning dado una métrica θ. Entonces pymach y sparkmach puede ser definido como un *pipeline* $q^*(G)$, donde

$$q^*(x) \in \arg\min_{\theta} \mathcal{L}(p(x))$$

$$\text{p(x)} = [\text{m}_{(x)}^{(0)}, m_{(x)}^{(1)}, \ldots, m_{(x)}^{(k-1)}] = \begin{cases} m^{(0)} & \text{1th input's mapper} \\ m^{(1)} & \text{2th input's mapper} \\ \vdots & \\ m^{(k-1)} & \text{kth input's mapper} \end{cases}$$

4.1.1. Estructura

Pymach está implementado en un formato modular, lo que permite la legibilidad, la depuración y la fácil interacción con otras aplicaciones. A continuación se detallará cada una de estas estapas.

4.1.2. Define (D)

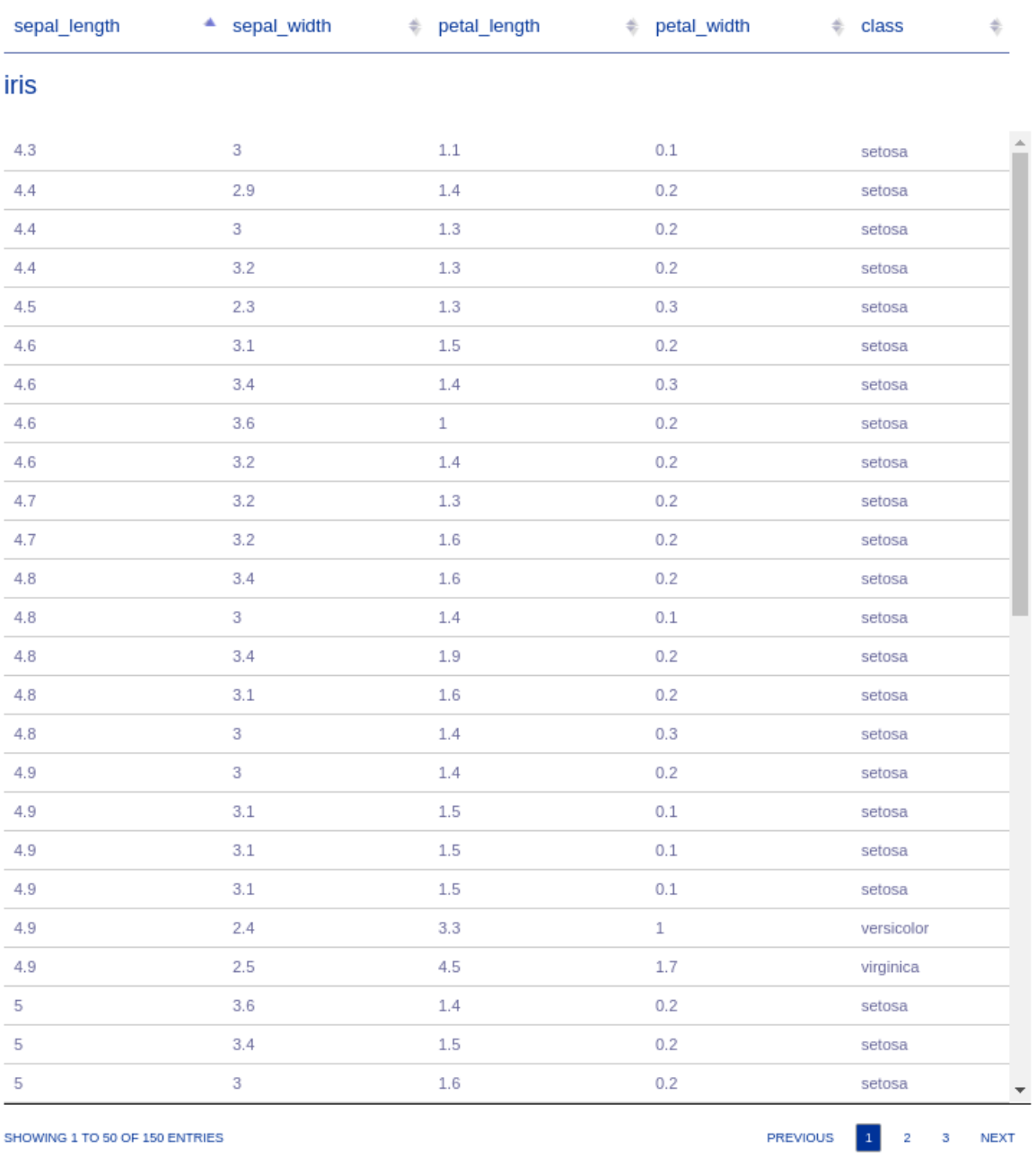
Este módulo se encarga de definir el conjunto de datos de entrada. Para la correcta definición se mide las siguientes características:

- **Ruta de archivo**: Ruta absoluta del archivo en formato CSV.
- **Cabecera** : Define los nombres de cada columna del archivo.
- **Respuesta** : Nombre de la columna a predecir.
- **Tipo de problema** : Indica si se trata de un problema de clasificación o regresión.
- **Número de características** : Define el número de variables predictoras, es decir, variables explanatorias.
- **Filas** : Define el número de filas
- **Tamaño** : Indica el tamaño del archivo en KB.
- **Data** : Objeto que contiene el dataset en un formato de DataFrame.
- **X** : Objeto que contiene, en formato de DataFrame, las columnas explanatorias.
- **y** : Objeto que contiene, en formato de DataFrame, la columna a predecir.

Dado estos parámetros se definen los caminos, DA, DP, y DS (ver figura 4.2). Los cuales son independientes entre sí y pueden ejecutarse en paralelo. La figura 5.3 muestra algunas caracteristicas extraídas de un conjunto de datos.

4.1.3. Analyze (A)

Este módulo se encarga de realizar un análisis exploratorio de los datos, es decir, mediante gráficas estadísticas, se obtiene una descripción de la data visualmente. Con esta información se logra percibir patrones en los datos, distribuciones, relaciones, entre



iris

sepal_length	sepal_width	petal_length	petal_width	class
4.3	3	1.1	0.1	setosa
4.4	2.9	1.4	0.2	setosa
4.4	3	1.3	0.2	setosa
4.4	3.2	1.3	0.2	setosa
4.5	2.3	1.3	0.3	setosa
4.6	3.1	1.5	0.2	setosa
4.6	3.4	1.4	0.3	setosa
4.6	3.6	1	0.2	setosa
4.6	3.2	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.7	3.2	1.6	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3	1.4	0.1	setosa
4.8	3.4	1.9	0.2	setosa
4.8	3.1	1.6	0.2	setosa
4.8	3	1.4	0.3	setosa
4.9	3	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
4.9	3.1	1.5	0.1	setosa
4.9	3.1	1.5	0.1	setosa
4.9	2.4	3.3	1	versicolor
4.9	2.5	4.5	1.7	virginica
5	3.6	1.4	0.2	setosa
5	3.4	1.5	0.2	setosa
5	3	1.6	0.2	setosa

SHOWING 1 TO 50 OF 150 ENTRIES PREVIOUS 1 2 3 NEXT

FIGURA 4.3: Conjunto de datos Iris, 4 caraterísticas (sepal length y width, petal length y width), 1 variable de respuesta (class), 150 filas, etc.

otros. A continuación se detalla cuatro gráficas principales en un análisis exploratorio convencional[28]. En la sección de experimentos y resultados se muestran ejemplos usando los diferentes gráficos.

- **Histograma**: Representa la distribución de frecuencias de las columnas. Si las variables son continuas se usan histogramas, en cambio, si son discretas o categóricas se usan diagramas de barras.

- **Diagrama de caja** : Se usa este gráfico para resumir la distribución de cada columna, dibujando una línea para la mediana, y una caja que abarca un 50 % de los datos, cuyos límites son el 25 percentil y el 75 percentil.
- **Matriz de correlación** : Es una gráfica que indica cuan relacionados estan dos variables. Si dos variables varían en la misma dirección, se dice que estan positivamente correlacionadas. Por el contrario, si varian en direcciones opuestas, se dice que estan negativamente correlacionados.
- **Matriz de dispersión** : Muestra la relación entre dos variables mediante puntos en 2 dimensiones. Atributos (características) que tengan una relación determinada (relación lineal, polinomial, no lineal, etc.) posiblemente estén correlacionadas, por lo tanto, serán buenos candidatos para ser removidos de los datos.

4.1.4. Prepare (P)

Este módulo se encarga de modificar la data para tener una data más limpia para el modelamiento. Algunas de estas técnicas empleadas son escalamiento y normalización, las cuales se encargan de tener cada elemento de la data en un determinado rango, con el objetivo de acelerar el modelamiento [29]. A continuación se presenta algunos métodos que ayudan a prepar la data para el procesamiento posterior [30].

- **Escalamiento Max y Min**: Escala cada valor numérico en el rango de 0 a 1. Los algoritmos que estan basados en gradiente se benefician mucho con este escalamiento.
- **Normalización** : Se reescala cada fila de la data, usando la norma l2, de tal manera que esa fila tenga longitud uno. Esto sirve para datos que contengan muchos valores ceros. Muy utiles para algoritmos basados en redes neuronales.
- **Escalamiento Estándar** : Escala los datos para que sigan una distribución normal de media 0 y desviacion 1. Este escalamiento es útil para algoritmos que supongan una distribución gausiana, como la regresión logística.
- **Escalamiento Robusto** : Es un tipo de escalamiento que es robusto a valores anómalos o atípicos. El escalamiento lo hace tomando en cuenta el primer y tercer cuartil.
- **Transformación de variables categóricas** : Es un método para transformar los valores categóricos de cada columna a valores numéricos.

4.1.5. Select (S)

Este módulo se encarga de realizar la selección de las mejores características de la data que mejoren el modelo final. Esta técnica tiene las siguientes funciones [28]:

- **Reducir el sobreajuste**: Al reducir la cantiadad de columnas, se reduce la complejidad del modelo, por tanto, se evita que el modelo prediga valores basado en ruido y pueda ser generalizado.
- **Mejora la precisión del modelo** : Al tener menos datos que no aporten al modelo, el modelo se entrena con valores menos redundantes, por ende, la precisión mejora.
- **Reduce el tiempo de entrenamieno** : Menos características, significa menos columnas para entrenar, por la tanto la matriz total se reduce considerablemente, disminuyendo el tiempo de entrenamiento.

En este módulo en particular se usan dos algoritmos para realizar la seleccion: Análisis de componentes principales (ACP, o en ingles, PCA) y un modelo ensamblado usando árboles de decisión aleatoria.

- **Análisis de componentes principales**: Este algoritmo se usa para reducir la dimensionalidad de la data (reducción de columnas). Una propiedad de esta técnica es que se puede escoger la dimensionalidad final deseada, comprimiendo toda la data a esa nueva dimensión.
- **Árboles de decisión aleatoria** : Este método ordena las características que aportan más al modelo. Una vez ordenado, se selecciona los mejores, reduciendo también la dimensionalidad. No es necesario escoger la dimensionalidad final.

4.1.6. Evaluate (E)

Este es el principal módulo que se encarga de aprender de los datos. En este módulo se realiza el paso de entrenamiento, el cual teniendo toda la data limpia y compacta, se ejecutan sobre ella diferentes algoritmos que tratan de obtener la mejor precisión en la prediccion.

La evaluacion de la data se han tomado en cuenta algoritmos tanto para clasificación o regresión, algunos de ellos se muestran a continuación:

- **Logistic Regression**: Es un algoritmo de clasificación binaria que usa la función logística o sigmoide, el cual es una curva que toma cualquier valor real y lo mapea a un valor entre 0 y 1, con el fin de predecir la probabilidad de que una observación (fila) caiga en uno de dos valores [7]. Este algoritmo tambien puede ser extendido para múltiples clases . Ver figura 4.4.

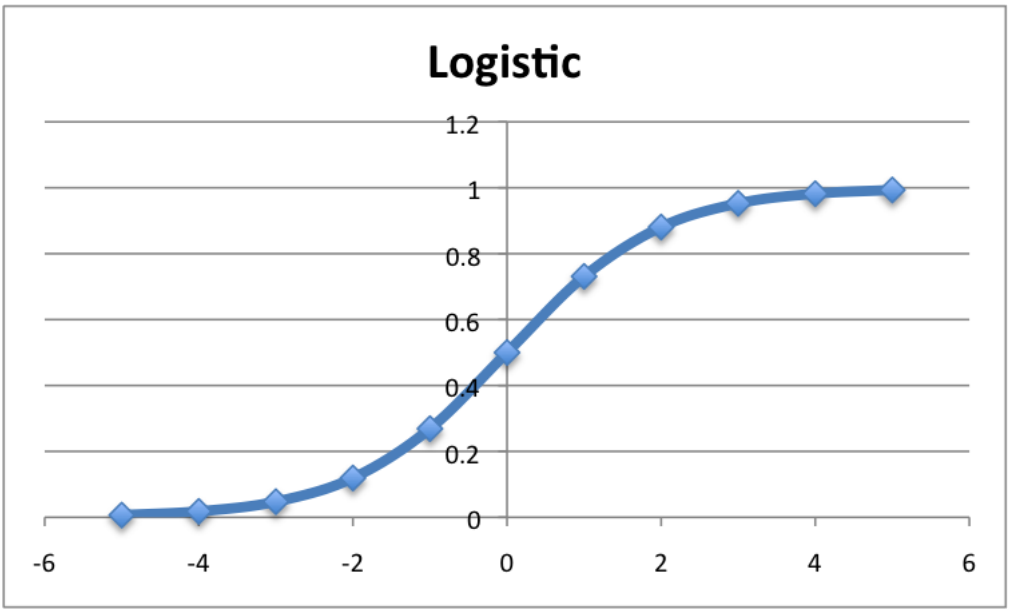


FIGURA 4.4: Funcion logística o sigmoide [7].

Consideraciones

- Este algoritmo es un algoritmo lineal, con función no lineal (sigmoide), por tanto, este asume una relación lineal entre las variables de entrada y la variable de salida (dependiente). Esto se logra transformando la data, con metodos de escalamiento, que es justamente un paso anterior a esta etapa.
- Este modelo puede tener sobreajuste (overfitting) si las variables de entrada estan correlacionadas. Esto se puede evitar eliminando variables que puedan ser expresadas por otra, previo diagnóstico usando el coeficiente de correlación de Pearson para variables numéricas.

- **Linear Discriminat Analysis** : Es un algoritmo diseñado para clasificaciones multiclase, tanto para dos o más clases, donde el objetivo principal es buscar un subespacio que maximice la separalidad de las clases [8], ver figura 4.5. Este algoritmo también puede ser usado para reducir la dimensionalidad de un conjunto de datos.

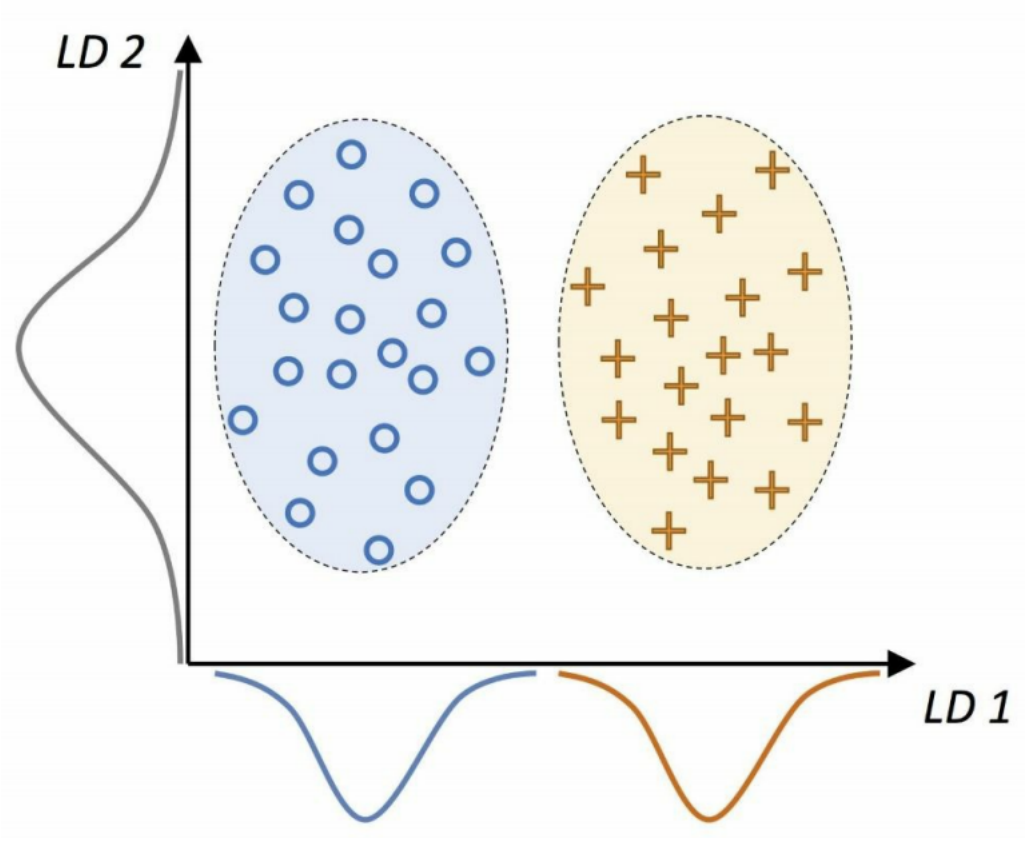


FIGURA 4.5: Se muestra dos clases bien separadas [8].

Consideraciones

- Se asume que la data tiene que ser normalmente distribuida.
- Sus matrices de covarianza tienen que ser idénticas y que cada variable de entrada debe ser independiente entre sí.

- **Gaussian Naive Bayes**: Naive Bayes es un algoritmo que tambien puede servir para problemas binarios y multiclase. Este clasificador está basado en el teorema de bayes, el cual calcula la probabilidad de que un particular objeto pertenezca a una clase determinada dado su conjunto de características [31]. La distribución que se escoja, recibe el nombre del algoritmo, puede ser una distribución de Gauss, Bernoulli, etc.

  Consideraciones

  - Se asume que la data tiene que seguir una distribución gaussiana.
  - Cada variable de entrada debe ser mutuamente independiente con las demás.

- **Multi-layer Perceptron Classifier** : Es un algoritmo diseñado para problemas lineales y no lineales, que aprende un óptimo conjunto de coeficientes que multiplicados con las variables de entrada predicen si un conjunto de datos pertenecen a

una clase o no. [9], ver figura 4.6. Para el caso de una versión multicapa, el número de capas ocultas puede ir creciendo, aumentando el nivel de complejidad de toda la red, lo que se conoce como aprendizaje profundo (deep learning), asimismo, el conjunto óptimo de coeficientes mejoraría en precisión a medida que se añaden más capas.

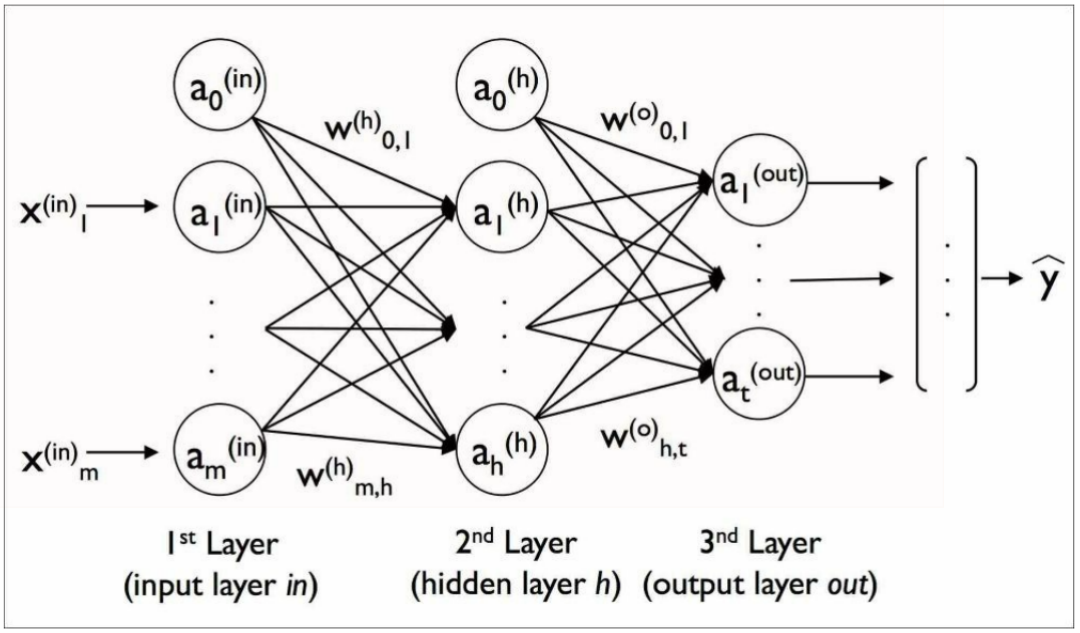


FIGURA 4.6: Ejemplo de una red neuronal multicapa [9].

Consideraciones

- El costo computacional para entrenar redes multicapa es alto.
- Funcionan muy bien para problemas tanto lineal y no linealmente separables.

- **SVC**: Este algoritmo es una extensión del algoritmo de máximo margen, el cual busca un hiperplano (separación) entre dos clases que maximice su separación, algo parecido a lo que buscada el algoritmo LDA, que en un lado del hiperplano se clasifique como una clase A, y en el otro lado del hiperplano se clasifique como una clase B. Lo añadido a este algoritmo es la creación de hiperplanos que puedan ser flexibles a errores, es decir, que un punto pueda pertenecer a un lado del hiperplano pero sin en realidad serlo. El otro añadido es no solo trabajar con hiperplanos, sino con hipersuperficies, lo cual es determinado por los kernels del algoritmo [10], ver figura 4.7.

Consideraciones

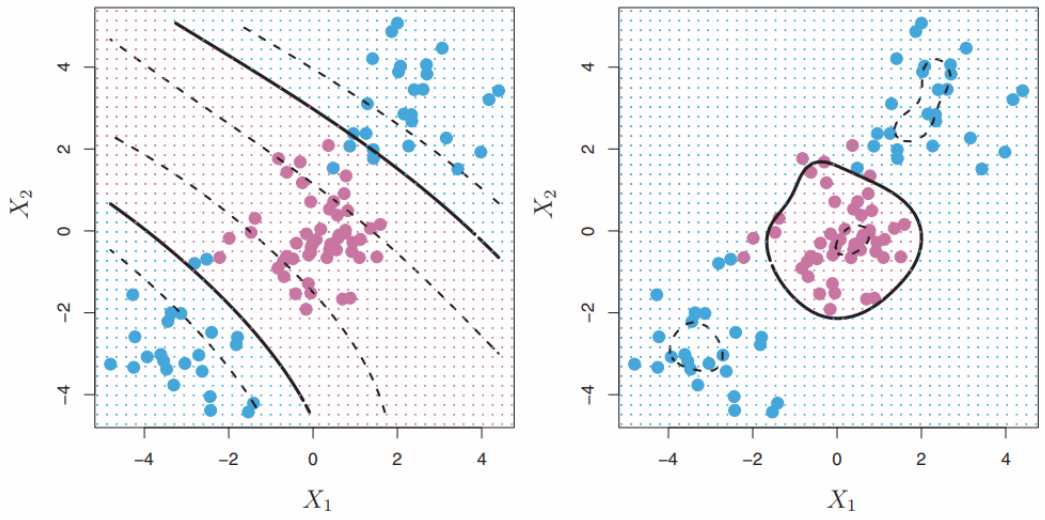


FIGURA 4.7: En la izquierda se muestra una frontera no lineal (hipersuperficie) de grado 3 (kernel polinomial) y a la derecha una frontera radial (kernel radial) [10].

- El algoritmo no requiere que los datos sean linealmente separables, es decir, la frontera que separe dos clases sea lineal, sino que puede funcionar también para casos no lineales.
- En el caso de problemas no linealmente separables, el costo computacional aumenta, esto debido a que el hiperplano creado tendria que ser de un orden superior.

- **DecisionTreeClassifier** : Este algoritmo busca segmentar el espacio de las variables de entradas mediante regiones, lo que se conoce como rectangulos de alta dimensión o cajas, de tal manera que un nuevo valor es filtrado a través del árbol y puesto en una de las regiones, el cual es el valor de salida del árbol [11], ver figura 4.9.

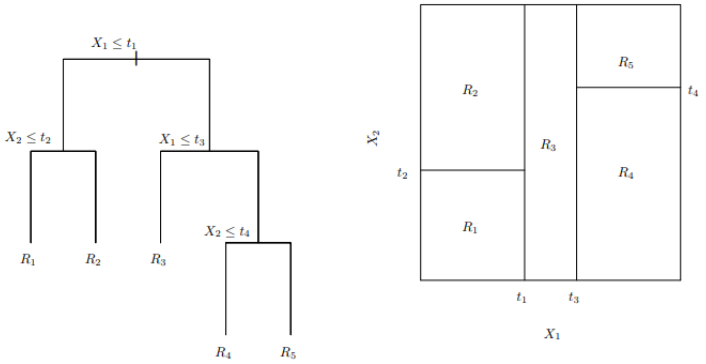


FIGURA 4.8: En la izquierda se muestra la regla de selección del árbol con dos variables y a la derecha las regiones creadas de predicción [11].

Consideraciones

- Los árboles de decisiones pueden aplicarse tanto para problemas de clasificación como de regresión.
- Los árboles son fáciles de explicar y graficar, lo cual facilita la interpretación a personas no expertas.
- Pueden manejar variables cualitativas sin crear variables dummy.
- No son tan robustos, ya que una variación en una observación, puede cambiar mucho la estimación del árbol. Sin embargo, usando métodos de Bagging y Boosting, el poder predictivo y la robustez aumentan.

Algunos otros modelos que pymach considera son: KNeighborsClassifier, RandomForestClassifier, ExtraTreesClassifier, GradientBoostingClassifier, AdaBoostClassifier y VotingClassifier.

4.1.7. Improve (I)

Una manera de mejorar el rendimiento de los modelos mostrados en la sección de Evaluate, es modificando los párametros por defecto de cada algoritmo. La técnica usada para tal función es la optimización de hiperparámetros, donde los parámetros de los algoritmos son llamados hiperparámetros. Hay principalmente dos métodos que se aplican para buscar el conjunto de valores (hiperparámetros) que optimicen el modelo [32], estos son:

- **Grid Search** : Técnica que modela y evalua un modelo para cada conjunto de valores entregados, esto lo hace de manera exhaustiva por fuerza bruta.
- **Random Search** : Parecido al Grid Search pero los valores entregados provienen de una muestra que son tomados de una distribución normal para un numero fijo de iteraciones.

Consideraciones

- El tiempo de la técnica Grid Search es muy alto dependiendo del conjunto de valores a entrenar, sin embargo, es más confiable ya que recorre todo la malla de valores.
- El Random Search, toma menos tiempo encontrando los hiperparámetros óptimos ya que solo toma una muestra de una distribución de parámetros.

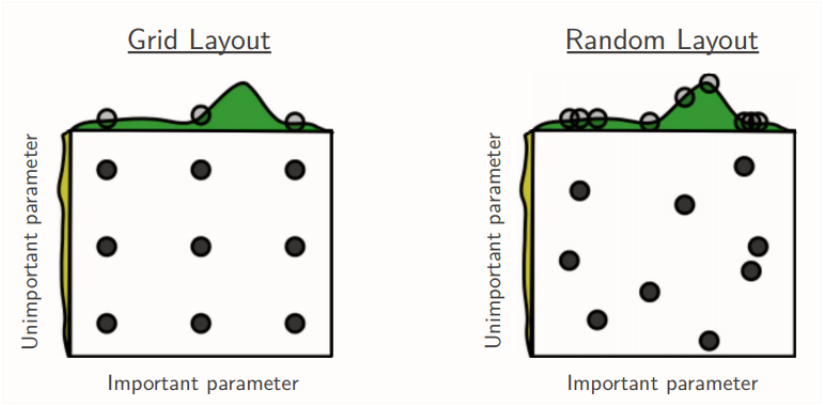


FIGURA 4.9: En la izquierda se muestra el Grid Search que solo toca en tres puntos a la función de objetivo, por el contrario, en la izquierda, el Random Search toca en los nueve puntos a dicha función, esto indica que la segunda tiene una mejor chance de encontrar una configuración óptima [12].

Sin embargo, no garantiza el conjunto de hiperparámetros óptimo para un conjunto de poca dimensión.

- Si el espacio de características (variables de entrada) es alto, la técnica por defecto seria el Random Search, ya que para una dimension alta, hay un conjunto de hiperparámetros que realimente importan del total, por lo tanto, probar con todos los valores (Grid Search) es muy ineficiente [12].

4.1.8. Present (P)

La presentación es un apartado muy importante en el flujo de un trabajo de ciencia de datos. En este apartado se muestran los resultados de una manera entendible y concisa, las cuales van a ser las gráficas que van a determinar la toma de decisiones.

Para graficar los resultados se ha hecho uso de la librería Plotly, la cual está escrita sobre D3.js, una de las librerías más populares escritas en JavaScript para la visualización de gráficos. Como se vió en el apartado de Analyze, para el analisis exploratorio se usó diferentes gráficas provenientes de estadística, como histogramas, diagramas de cajas, etc., sobre esa misma idea se muestran los resultados obtenidos por los modelos, esto es, los puntajes obtenidos y su varianza.

Capítulo 5

Sparkmach

5.1. Definición

Sparkmach es una librería escrita en Python[25] y Apache Spark[33] y a su vez, basada en Pymach.

En primera etapa se desarrolló una estructura especial llamada 'chain', el cual está basado en los pipelines de la libreria ML, cuya función es unir diferentes estimadores y transformadores. Esta implementación permite una mayor flexibilidad al trabajar con modelos, debido a que es fácilmente mantenible y escalable.

La diferencia entre Pymach y Sparkmach, es que el segundo está desarrollado para ser ejecutado en un cluster de computadoras ya que usa la técnica de *map reduce*, una técnica de paralelismo distribuido para grandes datos.

Algunas de las caracteristicas más importantes de esta librería son las siguientes:

- Al igual que Pymach, la ejecución de la librería es totalmente visual. La librería está desplegada en un servicio web, sin embargo, hace falta algunas configuraciones de instalación, ya que la filosofía de esta libreria es ejecutarse en un cluster de computadoras donde la data es alojada de manera distribuida en todos los nodos.
- Solo necesita la data y la variable que se quiere predecir, por ejemplo, un archivo en formato CSV y la columna de salida. Actualmente tiene soportado modelos de regresión.
- Es totalmente open source, con licencia MIT, lo que implica que puede ser copiado y modificado.

- Similar a pymach, la presentación de los resultados es interactiva, lo que facilita a la interpretación de los mismos.

Sparkmach esta dividido en 6 etapas (sin la etapa de Improve de Pymach), como se muestra en la figura 5.1, las cuales conforman un flujo de trabajo desde la entrada de datos hasta la presentación del dashboard de resultados. La elección de dicha arquitectura fue escogida con la finalidad de tener una librería altamente escalable y modular como Pymach, lo que implica que al actualizar un modulo este pueda mejorar el rendimiento de todo el conjunto y escalar horizontalmente al añadir más nodos. Los pasos son los siguientes:

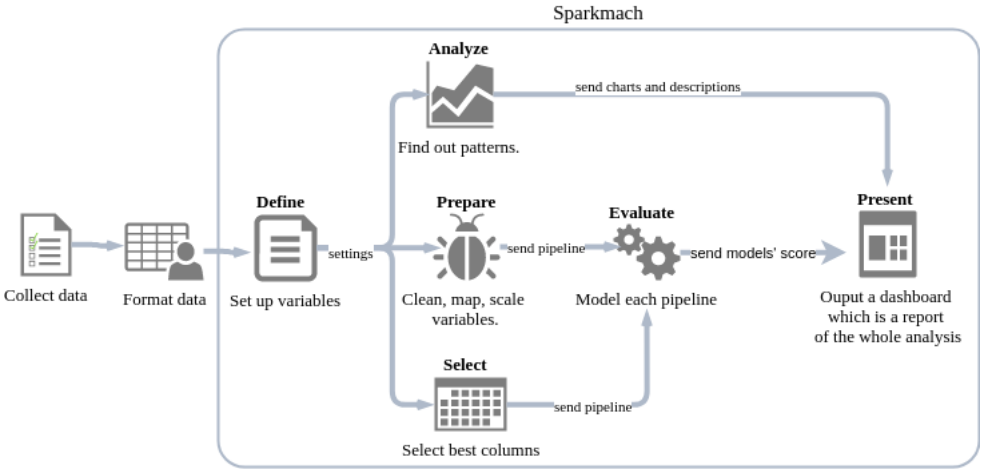


FIGURA 5.1: El flujo de trabajo de Sparkmach [13]. Va desde definir la data hasta la presentación de resultados.

- **Define (D)**: Definir el problema.
- **Analyze (A)** : Se encarga del análisis exploratorio de la datos.
- **Prepare (P)** : Modifica los datos.
- **Select (S)** : Escoge las mejores variables predictoras.
- **Evaluate (E)** : Ejecuta los modelos.
- **Present (P)** : Presenta el dashboard de resultados.

Al igual que Pymach todo esto se puede representar mediante un grafo de la siguiente manera:

Sea $G = (\hat{V}, \hat{E})$ un grafo dirigido o digrafo con un conjunto de vertices $\hat{V}$ y un conjuntos de aristas $\hat{E}$, de tal manera que $\hat{V} = \{D, A, P, S, E, I\}$, es decir, cada etapa es representado por un nodo. De la misma manera el conjunto $\hat{E}$ está conformado por los pares ordenados $\{(D, A), (D, P), (D, S), (A, P), (P, E)\}$ como se muestra en la figura 5.2.

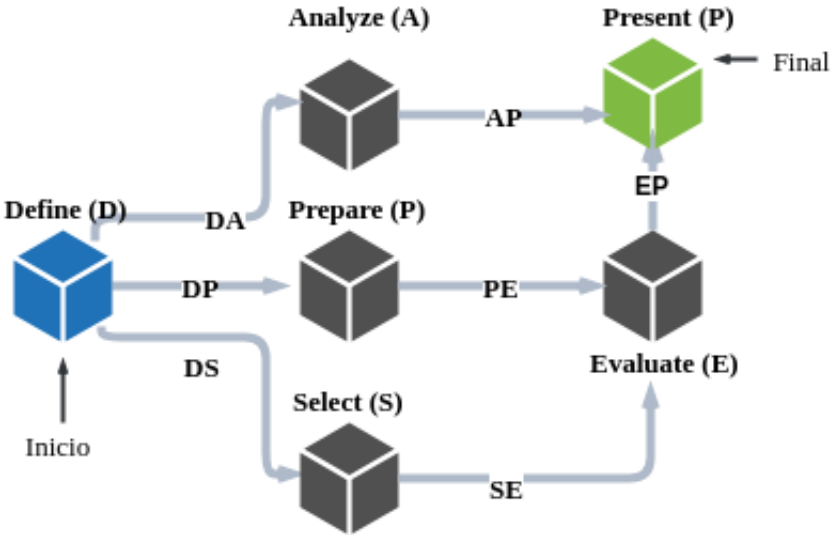


FIGURA 5.2: Representación de Sparkmach mediante un digrafo

5.2. Estructura

5.2.1. Define (D)

Sparkmach puede trabajar sobre diferentes versiones de un clúster, las probadas fueron YARN y Spark Standalone, donde la primera se encarga de asignar los recursos del sistema a la aplicación de manera automática, en cambio el modo Standalone requiere ingresar los recursos deseados, de memoria RAM y CPUs, para ejecutar una aplicación.

Para tal fin se trabajó con un archivo alojado en el HDFS de Hadoop, el cual permite que todos los nodos tengan acceso a él, acompañado de muchos más beneficios. A continuación se detalla las definiciones que se extraen del conjunto de datos:

- **Ruta del archivo**: Ruta absoluta del archivo alojado en el HDFS en formato CSV, TXT o PARQUET.
- **Cabecera** : Define los nombres de cada columna del archivo, puede dejarse vacio y tomar la cabecera del mismo archivo.

- **Respuesta** : Nombre de la columna a predecir.
- **Tipo de problema** : Indicar si se trata de un problema de clasificación o regresión (por defecto, está en regresión).
- **Numero de características** : Define el número de variables predictoras, es decir, variables explanatorias.
- **Filas** : Se establece el número de filas
- **Data** : Objeto que contiene el dataset en un formato de DataFrame de Spark.
- **X** : Objeto que contiene, en formato de DataFrame de Spark, las columnas explanatorias.
- **y** : Objeto que contiene, en formato de DataFrame de Spark, la columna a predecir.

Dado estos parámetros se definen los caminos, DA, DP, y DS (ver figura 5.2). Los cuales son independientes entre sí y pueden ejecutarse en paralelo. La figura 5.3 muestra algunas características extraídas de un conjunto de datos.

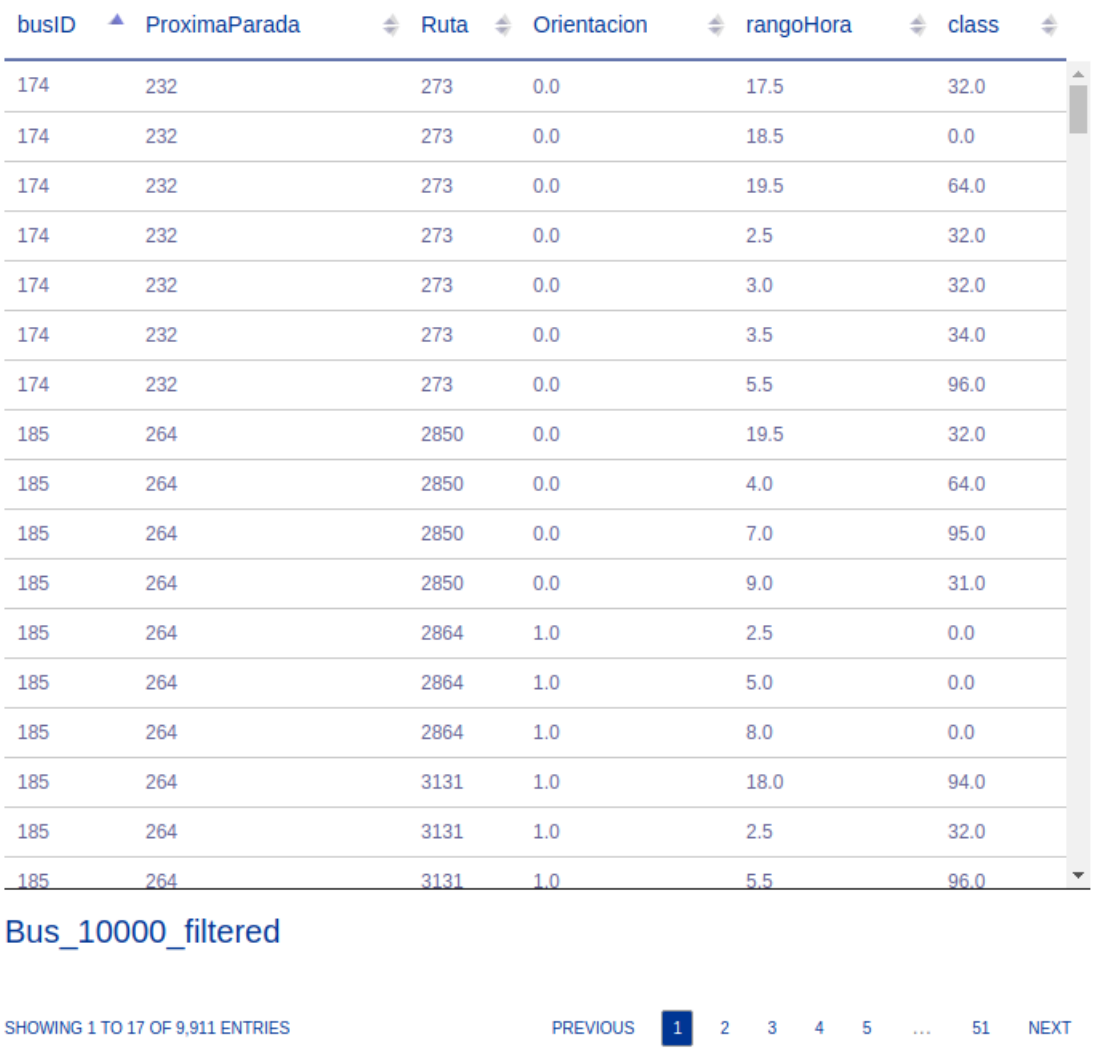
5.2.2. Analyze (A)

Este módulo es compartido con Pymach, ya que la generación de los graficos es generada del master del cluster, la cual no necesita ser paralelizada. El dataframe que contiene los resultados es un dataframe de tipo Spark, lo cual se transforma a una dataframe de tipo pandas, el cual es consumida con plotly y luego mostrada en un servicio web.

5.2.3. Prepare (P)

Los métodos de preparación de data son similares a los de Pymach, sin embargo, estos métodos requieren cambiar la estructura de datos a un vector de spark y luego ir aplicando el método a cada una de las variables de entrada. A continuación se presenta algunos métodos que ayudan a preparar los datos, basado en Pyspark [34].

- **Escalamiento Max y Min**: Escala cada dato numérico en el rango de 0 a 1. Dicha función espera un dataframe de spark.
- **Normalizacion** : Se reescala cada fila de la data, usando la norma l2, de tal manera que esa fila tenga longitud uno. Esto sirve para datos que contengan muchos valores ceros. Es recomendado en redes neuronales o para el algoritmo KNN.



busID	ProximaParada	Ruta	Orientacion	rangoHora	class
174	232	273	0.0	17.5	32.0
174	232	273	0.0	18.5	0.0
174	232	273	0.0	19.5	64.0
174	232	273	0.0	2.5	32.0
174	232	273	0.0	3.0	32.0
174	232	273	0.0	3.5	34.0
174	232	273	0.0	5.5	96.0
185	264	2850	0.0	19.5	32.0
185	264	2850	0.0	4.0	64.0
185	264	2850	0.0	7.0	95.0
185	264	2850	0.0	9.0	31.0
185	264	2864	1.0	2.5	0.0
185	264	2864	1.0	5.0	0.0
185	264	2864	1.0	8.0	0.0
185	264	3131	1.0	18.0	94.0
185	264	3131	1.0	2.5	32.0
185	264	3131	1.0	5.5	96.0

FIGURA 5.3: Dataset Bus (muestra de 10 mil filas), 5 caraterísticas (busID, Proxima-Parada, Ruta, Orientación, rangoHora), 1 variable de respuesta (class), 9911 filas, etc. Imagen de salida del software.

- **Escalamiento Estandar** : Escala los datos para que sigan una distribución Gausiana de media 0 y desviación 1. Este escalamiento es util para algoritmos que supongan una distribución gausiana, como la regresión logística o LDA. Funciona de la misma manera que el método de pymach, pero de manera distribuida.

5.2.4. Select (S)

Este módulo se encarga de realizar la selección de las mejores características de la data que mejoren el modelo final. Esta técnica tiene las siguientes funciones [28]:

- **Reducir el sobreajute**: Reduce la complejidad del modelo, por ende, mejora la predicción en casos no vistos.

- **Mejora la precisión del modelo** : Variables que esten fuertemente correlacionadas no aporta al modelo, la seleccion de caracteristicas evita la dependencia de variables.
- **Reduce el tiempo de entrenamieno** : Al tener menos variables explanatorias, el modelo requiere de menos recursos computacionales.

En este módulo se usa el selector chi cuadrado para seleccionar características en problemas de clasificación, para el caso de regresión todavia no está implementado:

- **Chi cuadrado**: Este algoritmo selecciona las mejores características en el cual la variable a predecir sea categórica, es decir, cuando se trate de un problema de clasificación.

5.2.5. Evaluate (E)

Este módulo evalua una lista de algoritmos en un conjunto de datos, similar a Pymach. Con la diferencia, mencionado antes, de que todo el código es reescrito para evaluar modelos para grandes datos. Algunos de los modelos similares a Pymach son:

- LinearRegression: Regresión lineal clásica que asume una distribución normal. Similar al que ejecuta Pymach.
- DecisionTreeRegressor: Es un árbol de decisión aplicado a la regresión. Similar al de Pymach.
- RandomForestRegressor: Es un conjunto de arboles de decisión que en conjunto mejora la predicción de un solo árbol.
- GBTRegressor: Es un tipo de algoritmo de regresión más robusta y que es fácilmente paralelizable, basado en optimización por gradiente.

Y algunos propios de Spark:

- GeneralizedLinearRegression: Al contrario a la regresión típica, que asume que la distribución de los datos de entrada siguen una normal, este algoritmo asume alguna distribución de una familia de los exponenciales [35].
- AFTSurvivalRegression : Es usado para modelar el deterioro de ciertos sistemas, biológicos o mecánicos. Generalmente la variable de respuesta es el tiempo, en el cual el sistema entrará en un estados de no actividad [36] [37].

Capítulo 6

Experimentos y resultados

Se probó tanto Pymach como Sparkmach con 4 tipos de datos, *Iris*, *Buses de New York*, un conjunto de datos de una simulación de la avenida Tupac Amaru, Lima, Perú y una simulación física. Cada conjunto de datos pasó por cada uno de las etapas que comparten Pymach y Sparkmach, pero con dimensionalidades distintas, Pymach se probó con la data de Iris y la data simulada que tienen un número de filas y columnas manejables, en cambio, la data de New York y la simulación física que tienen millones de registros y muchas columnas se probó usando Sparkmach para visualizar la escalabilidad del mismo.

6.0.1. Conjunto de datos : Iris

El conjunto de datos Iris fue presentado por Ronald Fisher en su paper *The use of multiple measurements in taxonomic problems* en el año 1936. Este conjunto de datos mide el largo y ancho de sépalos y pétalos de 3 tipos de flores: Iris setosa, Iris virginica e Iris versicolor. A continuación de muestran los resultados.

- **Histograma**: Se observa en la figura 6.1 que todas las variables de entrada se aproximan a una distribución casi normal, sin contar la variable de salida, *class*, el cual es la clase de flor a predecir. Sabiendo que las variables de entrada siguen esta distribución, es más fácil para un modelo tener los datos con esta tendencia, ya que la mayoria de los modelos de aprendizaje funcionan muy bien con el comportamiento de estos datos, por ejemplo, la regresión logística, supone que los datos siguen esta distribución para optimizar la efectividad del modelo.

- **Diagrama de caja** : En la figura 6.2 se puede observar que 3 de las 4 variables no tienen valores atípicos, sin embargo, la variable *sepal_width* tiene valores anómalos.

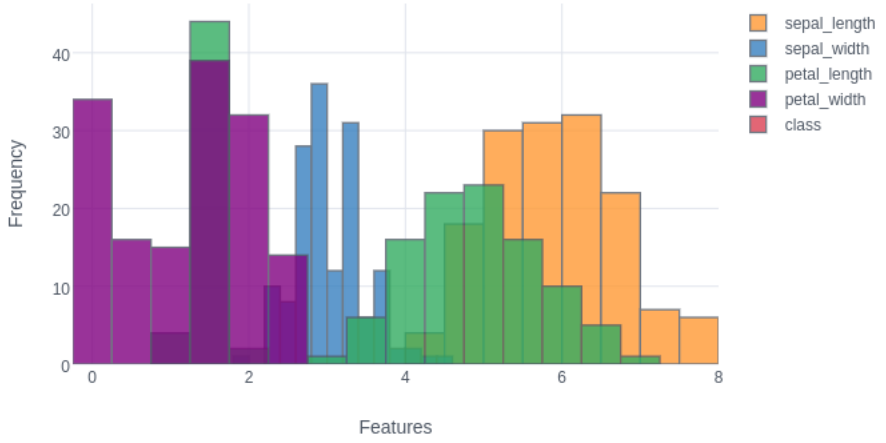


FIGURA 6.1: Histograma para el conjunto de datos Iris.

Este análisis es importante para ver que está pasando con estos puntos, cual es su comportamiento y en qué afecta a las demás variables y al modelo en general.

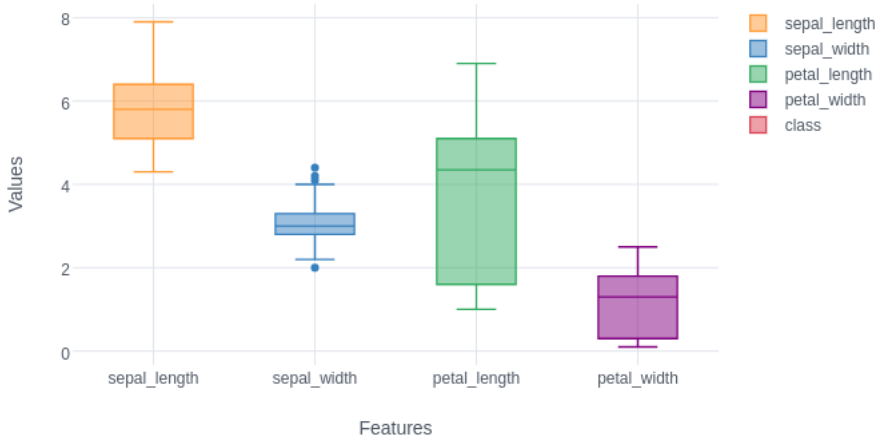


FIGURA 6.2: Diagrama de caja o Boxplot para el conjunto de datos Iris.

- **Matriz de correlación** : Se puede observar en la figura 6.3 que las variables *petal_width* y *sepal_width* presentan un alto grado de correlación (cercana a 1), indicando que una variable puede ser expresada como la otra, en otras palabras, se pudiera reducir las dos variables a una sola mediante un operación de combinación

o eliminación. En cambio la correlación entre las variables *sepal_width* y *petal_length* es muy baja, cercana a -0.4, lo que indica poca correlación y relación negativa.

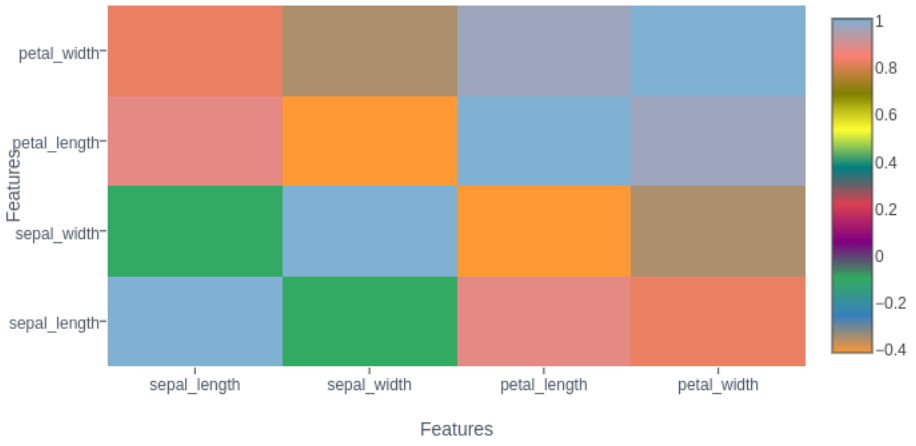


FIGURA 6.3: Matriz de correlación para el conjunto de datos Iris.

- **Matriz de dispersión** : Otra forma de poder ver relaciones entre las variables es mediante este tipo de gráfico, por ejemplo, se observa que *sepal_length* y *petal_length* forman dos agrupaciones, indicando una posible clasificación . Ver figura 6.4.

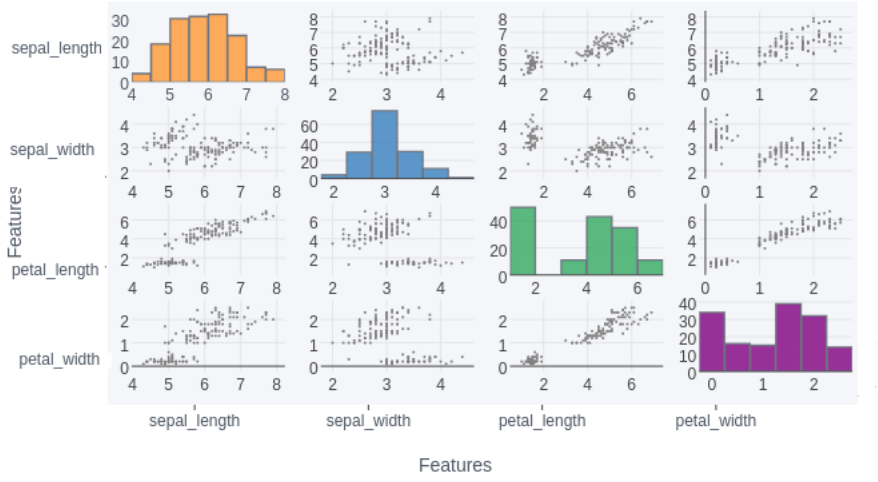


FIGURA 6.4: Matriz de dispersion para el conjunto de datos Iris.

Model Report

Models' report

MODEL	MEAN	STD
GaussianNB	0.9790000000000001	0.033
LogisticRegression	0.9790000000000001	0.033
LDA	0.972	0.034
KNeighborsClassifier	0.972	0.047
RandomForestClassifier	0.972	0.034
SVC	0.965	0.048
ExtraTreesClassifier	0.958	0.047
Voting(GBC-ET)	0.958	0.047
GradientBoostingClassifier	0.9570000000000001	0.057
AdaBoostClassifier	0.93	0.069
DecisionTreeClassifier	0.929	0.055

FIGURA 6.5: De arriba hacia abajo se muestra de manera ordenada los modelos entrenados para un conjunto de datos. La ordenación es de forma descendente, el primero es el mejor modelo y el último el peor.

Asimismo, la misma tabla se muestra en un diagrama de caja, para visualizar bien la distribución de los puntajes, y comparar entre el mejor y peor modelo. Ver figura 6.6.

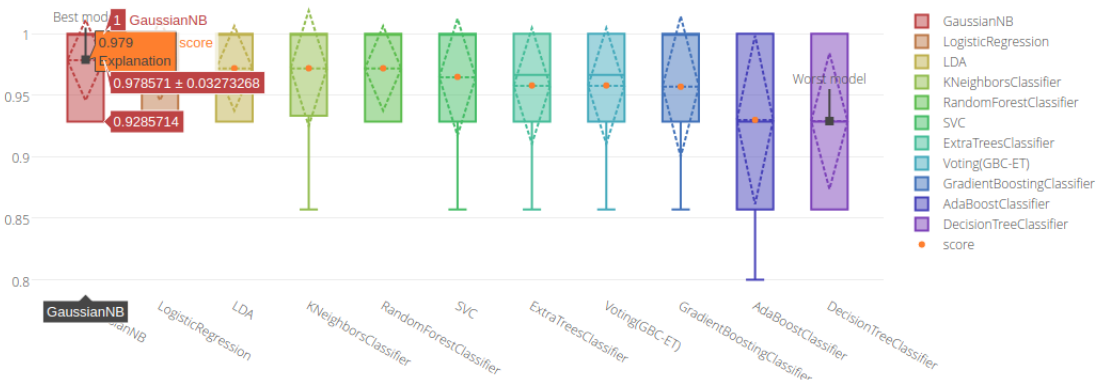


FIGURA 6.6: De izquierda a derecha se observa la misma ordenación descendente, donde el modelo más a la izquierda es el mejor, y en el otro extremo de la izquierda se encuentra el peor. Además, se logra ver la distribución de los puntajes obtenidos en la validación cruzada, sus extremos y su mediana.

6.0.2. Conjunto de datos : Data simulada de la av. Tupac Amaru

Simular parte de la avenida Tupac Amaru parte de la idea de ver como se comporta Pymach y Sparkmach para datos que podrian provenir de sensores y ver su comportamiento en problemas relacionados al tráfico vehicular.

Para tal fin, se escogió un algoritmo perteneciente a la teória de automátas celulares, el cual está basado en la discretización del espacio en celdas, que son la analogía de los vehículos, y gobernado por una regla de formación local [38]. En otras palabras, un automáta celular es un conjunto de celdas en una matriz, de tal modo que el estado de cada celda varía dependiendo a la regla que se le haya establecido[39]. La figura 6.7 muestra la vecindad de una celda en una grilla o matriz.

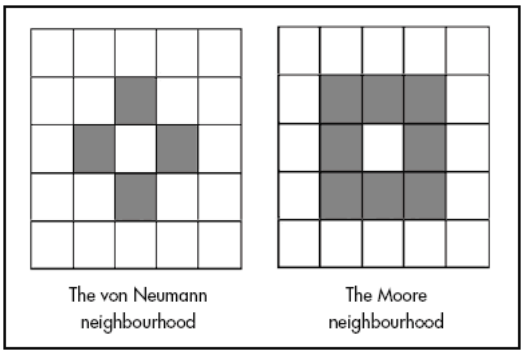


FIGURA 6.7: Estado de las celdas vecinas según el tipo de vecindad [14].

Basado en esta teoría, se hace uso del modelo Nagel-Schreckenberg, el cual, simula el flujo de vehículos en un array unidimensional (escalable a multidimensional) donde cada celda representa un segmento de la pista y el estado de cada celda depende de una regla (función de transición) [40]. Estas reglas se pueden observar en la figura 6.8

La elección de los parámetros de entrada de la simulación, está basada en un tramo de la avenida Tupac Amaru del distrito del Rímac, Lima, Perú. La figura 6.9 muestra el mapa (Google Maps) de dicho tramo, el cual empieza en el cruce de la av. Tupac Amaru con Caquetá hasta la av. Tupac Amaru con la av. 18 de Enero (Metro UNI). Los parámetros obtenidos de dicho tramos fueron:

- **Nivel de tráfico en diferentes segmentos dentro del tramo**

  Aquí se observa los segmentos de color celeste y amarillo, el segmento celeste indica tráfico fluido y el amarillo tráfico moderado, del cual se extrajo la velocidad media en dichos tramos.

- **Distancia entre segmentos**

  En la misma figura 6.9 se observa un segundo mapa que indica las distancias entre dichos segmentos, con esto se logró mapear dicha topología en la simulación. Así como también, el número de carriles.

Semáforos

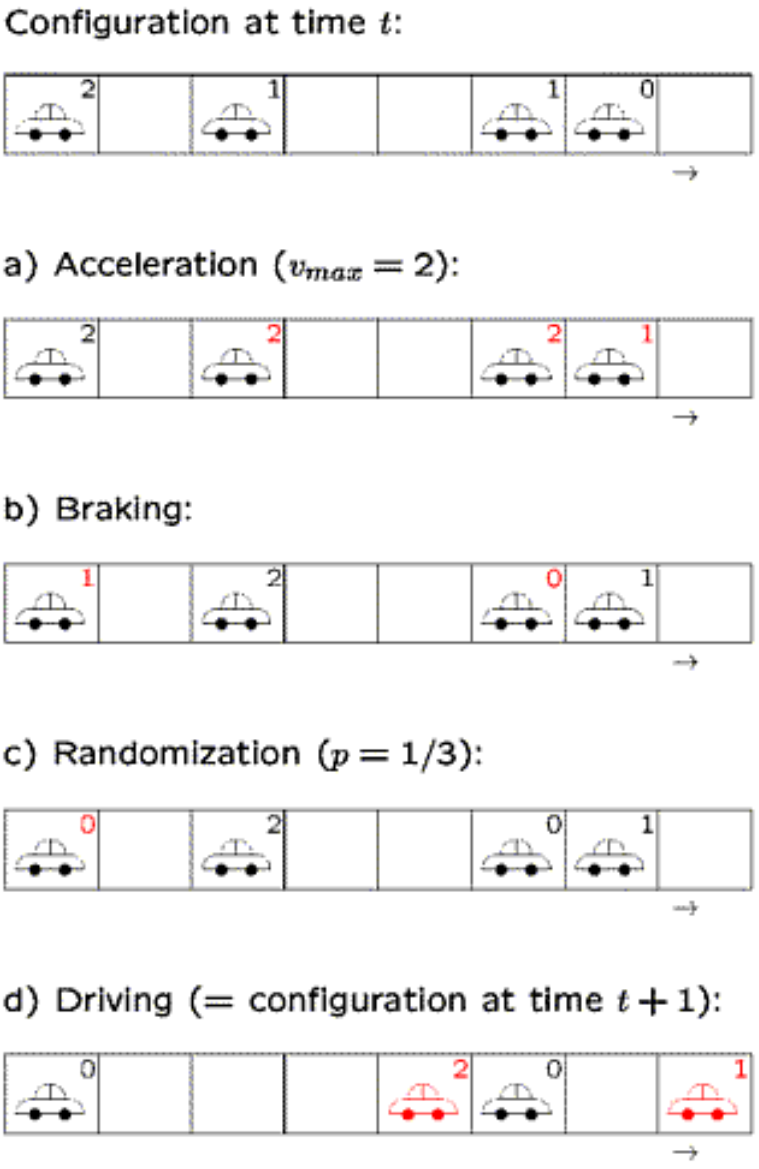


FIGURA 6.8: Función de transición en el modelo N-S [15].

Se implementó también las posiciones de los semáforos, en este caso, fueron dos: En el cruce con la av. Eduardo de Habich y otro en el cruce con la av. Honorio Delgado.

En total se obtuvo los siguientes parámetros:

- Carriles = 4
- Longitud = 2220 m
- Velocidad media = 33 km/h (dado la distancia y el tiempo mostrados en el mapa)

TABLA 6.1: Salida de la simulación.

deadCars	flux	time	avgSpeedCars	density	totalCars
31	16.82	101	2.11	7.95	70
33	15.45	102	2.00	7.73	68
35	15.23	103	2.03	7.50	66
37	16.70	104	2.19	7.61	67

- Semáforos = 2
- Límites de velocidad : Fueron dos, en los rangos [584m, 742m] y [1720m, 1770m]

Otros parámetros que fueron asumidos fueron:

- Carros que ingresan = [0, 2], pueden ser 0, 1 ó 2 carros que ingresan en el inicio del tramo.
- Velocidad máxima = 5*9 = 45 km/h
- Velocidad moderada = 3*9 = 27 km/h
- Probabilidad de frenado = 0.3
- Probabilidad de cambio de carril = 0.2

Todos estos parámetros de entrada fueron considerados para mapearlos en la simulación, como se muestra en la figura 6.10, en donde los cuadrados rojos indican vehículos con velocidad cero y los cuadrados verdes, los vehículos con velocidad máxima; los colores intermedios varían entre la velocidad máxima y mínima.

Se logró obtener 6 valores de dicha simulación, **deadCars, flux, time, totalCars, density** y **avgSpeedCars**. Estas 6 variables serán usadas para crear un modelo predictivo, 5 de ellas son las características (deadCars, flux, time, totalCars, density) y 1 (avgSpeedCars) fue la variable a predecir. Como se muestra en la figura 6.1.

- **Histograma**: Se observa en la figura 6.11 dos distribuciones, la de flujo (*flux*) y la de velocidad (*class*), la primera tiene una distribución casi normal, sin embargo, la segunda tiene un sesgo hacia la izquierda, indicando que a más tiempo, la velocidad iba disminuyendo.

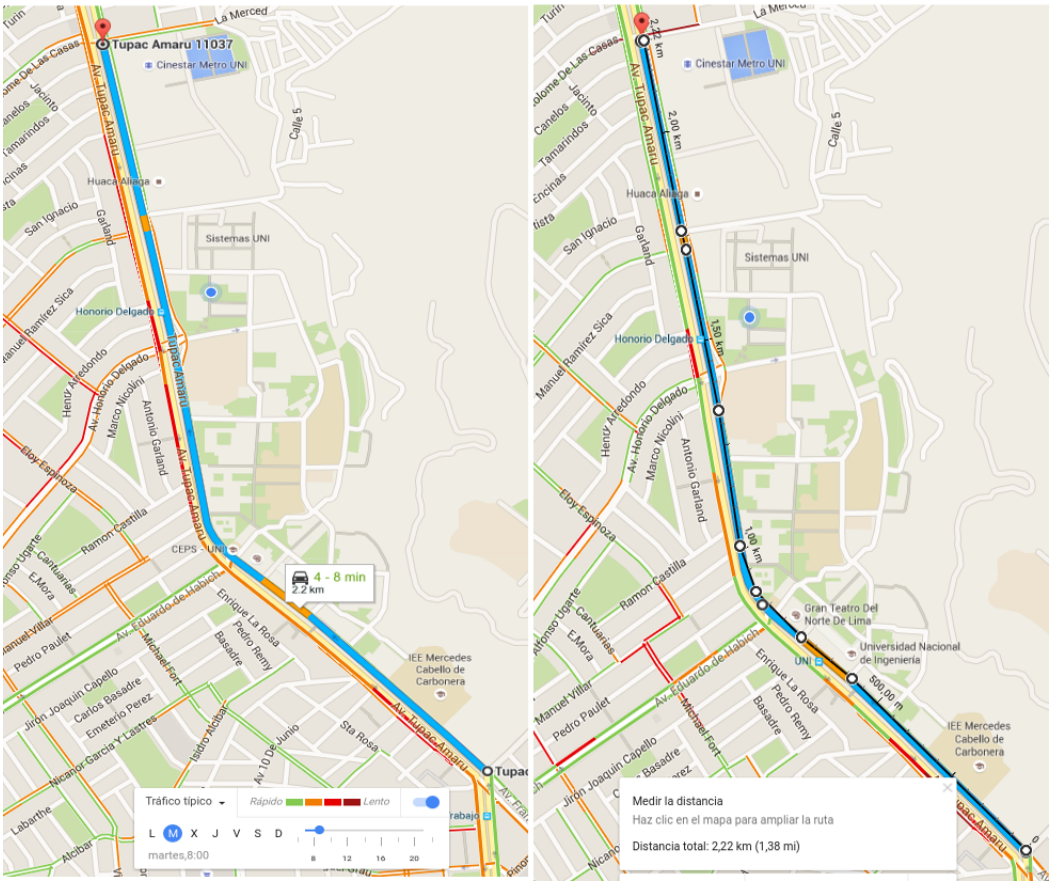


FIGURA 6.9: Mapa del tramo de la avenida Tupac Amaru, según Google Maps.

FIGURA 6.10: Tramo en el simulador.

- **Diagrama de caja** : En la figura 6.12 se puede observar que la variable *queue* tiene una distribución más conglomerada hacia el valor 0, esto indicando que al inicio de la simulación, no hubo ningún carro en cola.

- **Matriz de correlación** : Se puede observar en la figura 6.13 altas correlaciones entre las variables [flux, avgSpeedCars], [density, avgSpeedCars] y [totalCars, avgSpeedCars] y bajas correlaciones entre [deadCars, avgSpeedCars] y [time, avgSpeedCars], con estos resultados se logra comprender que las variables **flux, density** y **totalCars** aportan mucho al modelo, en cambio **deadCars** y **time** en poca medida. Sin embargo, el tiempo es una variable muy importante en todos los modelos,

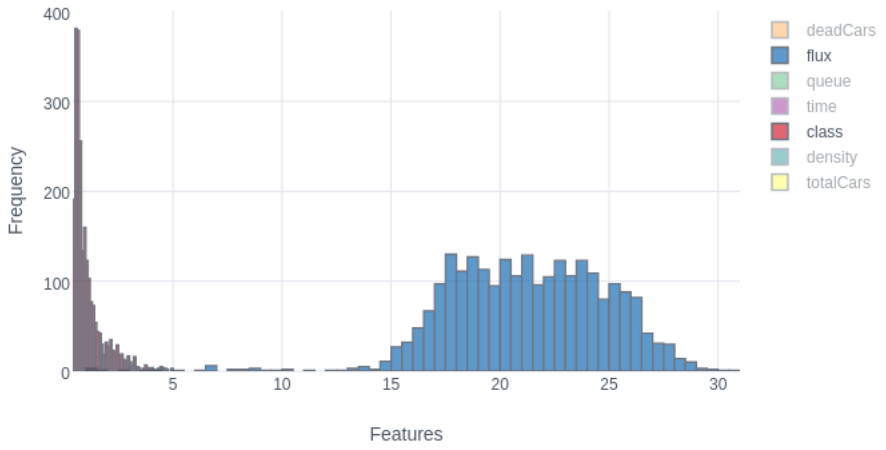


FIGURA 6.11: Histograma para la data simulada de la av. Tupac Amaru

FIGURA 6.12: Diagrama de caja o Boxplot para la data simulada de la av. Tupac Amaru.

ya que es una variable inherente en todo fenómeno físico, por ello, estuvo presente en la creación del modelo predictivo, que se detallará más adelante. Ahora viendo un análisis entre las características (todas menos avgSpeedCars) se logra ver, por ejemplo, [density, flux], [flux, totalCars], [time, flux], entre otras, que tienen baja correlación, es decir, son independientes entre sí, pero la relación [density, totalCars] tiene alta correlación, por tanto, dependientes entre sí, por lo tanto, estas variables podrían fácilmente convertirse en una sola, teniendo ahora solo 5 características y por lo tanto mejorando los tiempos de cálculo en la creación del modelo (se pasaría de una matriz de Nx6 a una de Nx5).

- **Matriz de dispersión** : En la figura 6.14 las relaciones entre dos variables, por ejemplo, *queue* y *deadCars* tienen una relación directa positiva.

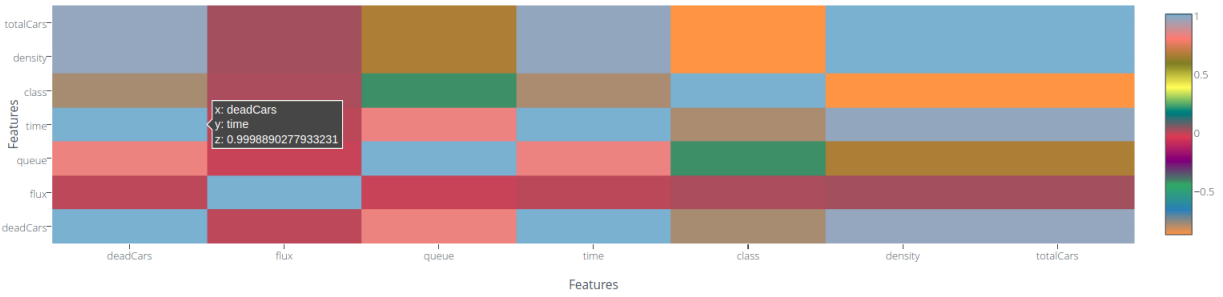


FIGURA 6.13: Matriz de correlación para la data simulada de la av. Tupac Amarus.

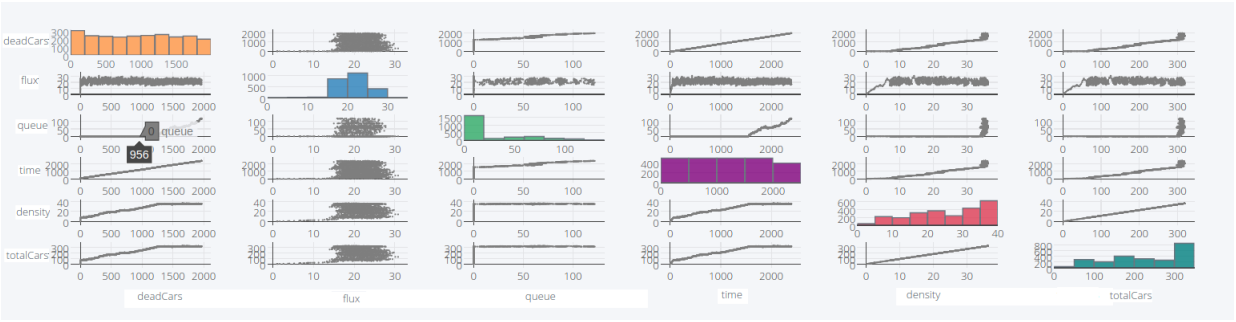


FIGURA 6.14: Matriz de dispersión para la data simulada de la av. Tupac Amaru.

- **Modelamiento Pymach y Sparkmach** : Se ha realizado una comparación entre los resultados que arrojan Pymach y Sparkmach para el modelamiento, midiendo puntajes de los modelos y los tiempos de ejecución. En la figura 6.15 se muestra una comparación de puntajes entre pymach y sparkmach, se observa que ambos tienen puntajes similares, con ligera mejoría en Pymach. Sin embargo,se observa también que los tiempos son menores en Pymach que Sparkmach. Esto se debe básicamente a que la data no es lo suficientemente grande como para que los tiempos de comunicación entre nodos sea mucho menor al tiempo de ejecución de los algoritmos; en otra palabras, la comunicacion entre los nodos al aplicar el paradigma *map reduce* es un cuello de botella en comparación al tiempo de demora de los modelos, debido a que los bloques que se mapean en los nodos son muy pequeños.

6.0.3. Conjunto de datos : Datos de buses de Nueva York

Para probar Sparkmach de manera más eficiente, se trabajó con una data más pesada en términos de dimensión. Para tal fin, se obtuvo data de los buses de New York [41], básicamente de sensores que miden la posición de los buses, su recorrido, velocidad y entre otras características más. La tabla 6.2 muestra las variables a considerar. Este

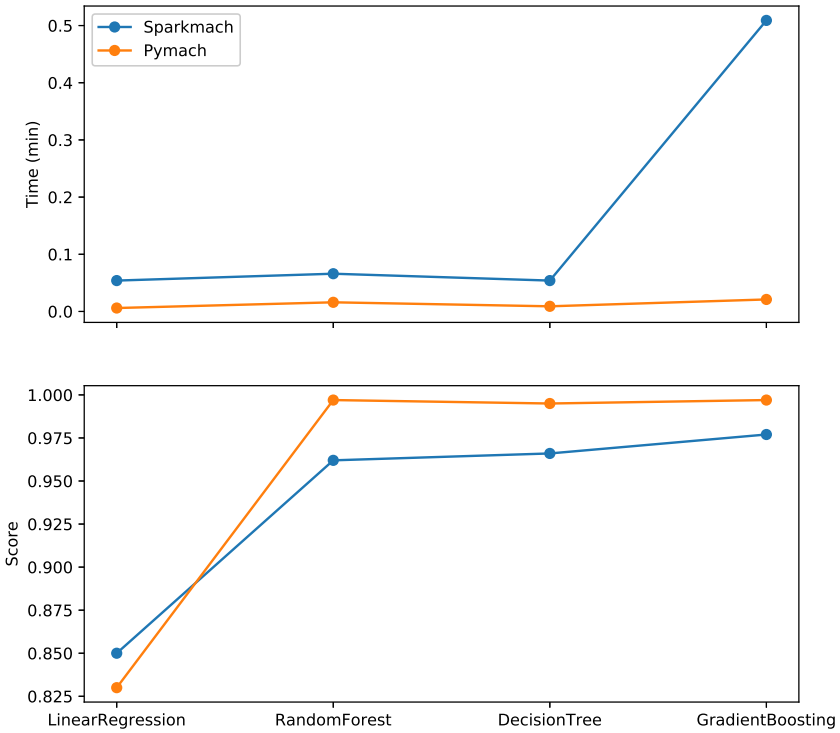


FIGURA 6.15: Comparación entre Pymach y Sparkmach para la data de simulación.

TABLA 6.2: Conjunto de datos buses de Nueva York.

BusID	NextStop	Route	Orientation	HourBucket	response
174	232	273	0	17.5	32
174	232	273	0	18.5	0
174	232	273	0	19.5	64
174	232	273	0	2.5	32

apartado se centra en ver como se da el escalamiento de sparkmach en un clúster de computadoras para dos muestras. Esta comparación se puede observar en la figura 6.16.

La figura 6.16 muestra claramente que a mayor cantidad de data el escalamiento es más considerable, ya que soluciones de Big Data funcionan mejor con datos masivos, debido a que la comunicación entre nodos es menos considerable a la cantidad de datos que fluyen entre ellos. Asimismo, la curva de escalabilidad (curva decreciente) es notoria a medida que los nodos aumentan. Esta escalabilidad se debe básicamente a que Spark realiza un paradigma basado en Map Reduce pero en memoria RAM. Adicionalmente, Spark introduce una principal abstración para trabajar con data, la RDD (Resilient

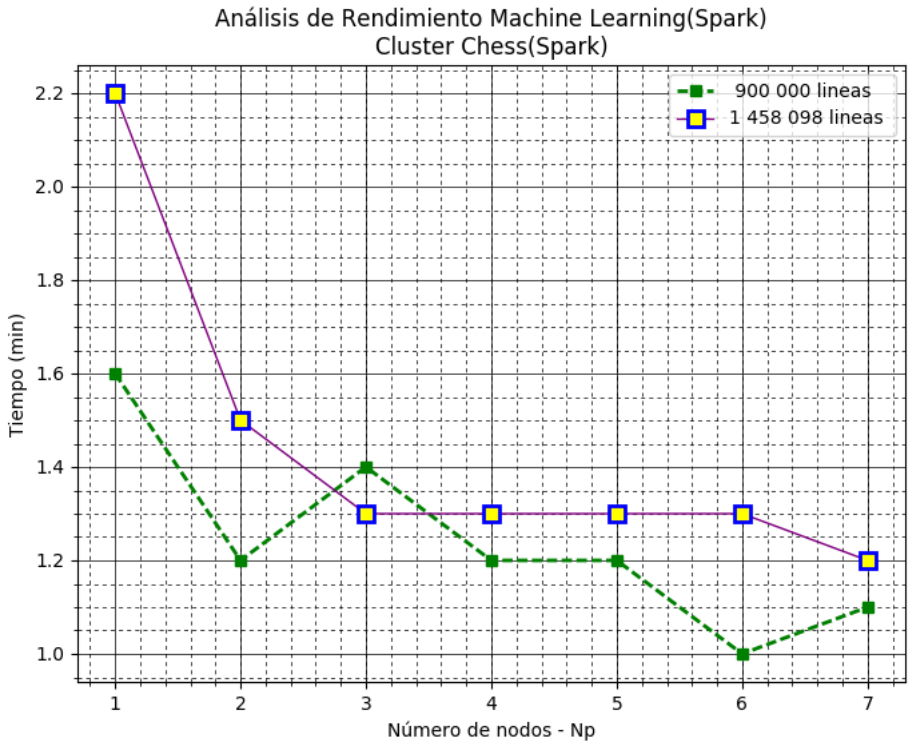


FIGURA 6.16: Prueba del datasets de bus en un clúster de 7 nodos.

Distributed Dataset) los cuales son una colección de objetos distribuidas, donde cada RDD es dividida en multiples particiones, cuyas particiones pueden ser computadas en diferentes nodos de un clúster [33], como se observa en la figura 6.17.

Para dar una visión más clara de como funciona estos algoritmos, supongamos que se quiere paralelizar el algoritmo del Gradiente Descendente aplicado a una regresión lineal o a una regresión logística, entonces los pasos a realizar para paralelizar dicho algoritmo son los siguientes:

En la figura 6.18 se muestra los w_i los cuales tienen que ser actualizados en cada iteración hasta converger, de tal manera que se tenga todos los pesos (parámetros del modelo) para crear la función de hipótesis, la cual será usada para hacer las predicciones. Entonces, con dicho objetivo, se realizó lo siguiente: se dividió la data y se repartió a los 'workers', cada 'worker' realiza ciertas multiplicaciones de matrices, y finalmente estos resultados se suman, y se le asigna a un parámetro w_i.

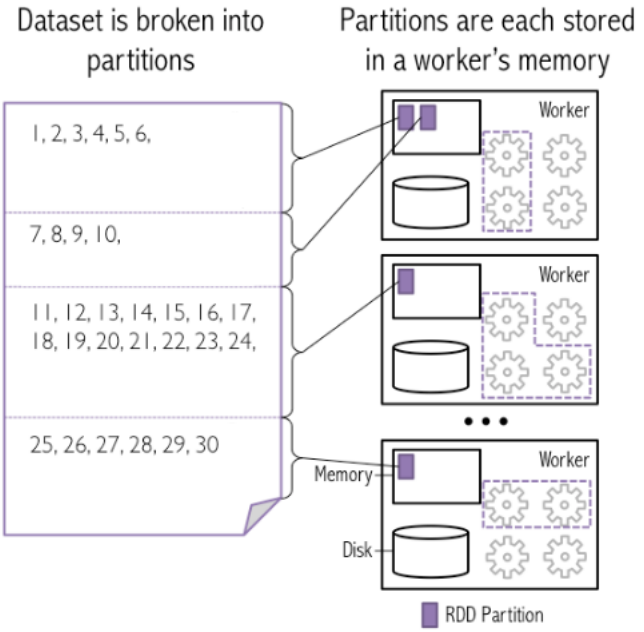


FIGURA 6.17: Particionamiento de la data en un paradigma *Map Reduce* [16].

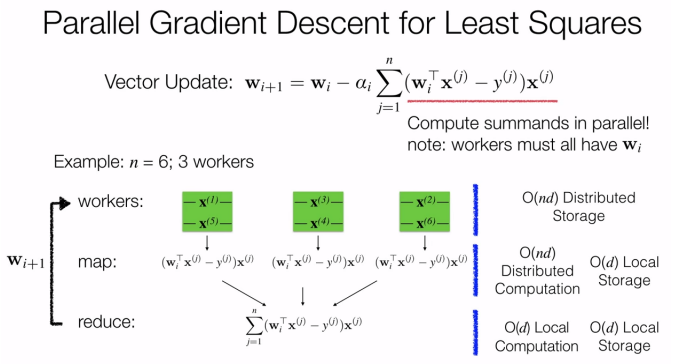


FIGURA 6.18: Algoritmo de Gradiente Descendente en paralelo [17].

6.0.4. Conjunto de datos : Experimento en física de altas energías

Finalmente, se probó Sparkmach con un conjunto de datos proveniente de un conjunto de simulaciones, usando el método Monte Carlo, para medir las colisiones que producen ciertas particulas elementales, asimismo, medir el decaimiento de sus productos [42]. La tabla 6.3 muestra las variables a considerar.

Se probaron dos algoritmos de clasificacion: *Logistic Regression* y *Random Forest* tanto en Pymach como en Sparkmach. La figura 6.19 muestra los resultados.

Estos resultados reflejan claramente la escalabilidad de Sparkmach manteniendo la predictibilidad casi sin variar. Los tiempos en Pymach tienen un comportamiento polinomial, mientras que en Sparkmach estos son lineales. Lo màs interesante, es que usando

TABLA 6.3: Datos de simulación HEPMASS. Esta data cuenta con 27 columnas y 10.5 millones de filas.

f_1	f_2	f_3	...	f_27	response
1.816216	-0.458454	1.010234	...	0.977542	1
-0.196046	-0.737780	-0.164820	...	-1.448544	0
1.235821	-0.970305	-1.456574	...	1.414154	1
-0.541583	1.284816	-0.645087	...	0.090727	0
0.594807	1.535389	-0.419327	...	0.666845	1

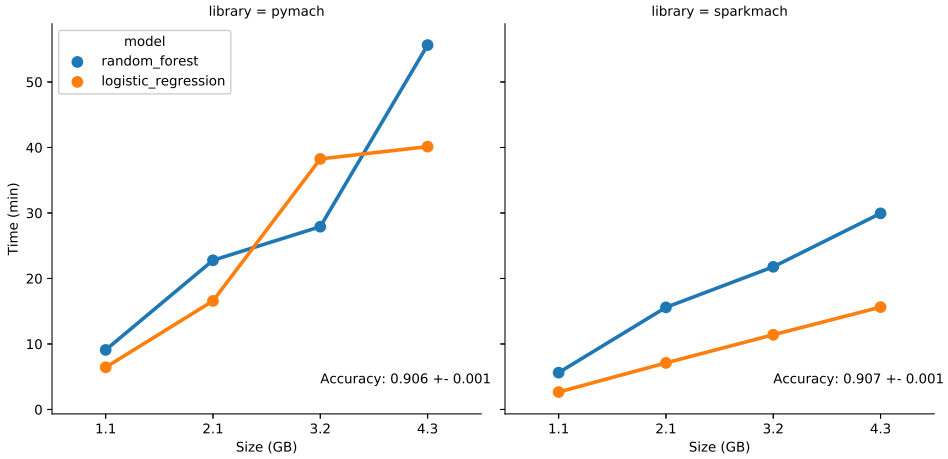


FIGURA 6.19: Prueba de escalibilidad y predictibilidad para la data HEPMASS.

pymach o sparkmach, se pueda lograr una predictibilidad razonable solo con algoritmos base, es decir, con algoritmos a los que no se le han dedicado mucho tiempo en su afinamiento.

Capítulo 7

Conclusiones y propuestas

7.1. Conclusiones

La automatización del proceso de ciencia de datos involucra no solo una metodología secuencial de pasos que ataquen un problema específico en todo el flujo (exploración, preparación, modelamiento, optimización), sino también involucra un conocimiento del negocio de la data a tratar, por lo tanto, tanto pymach como sparkmach son una base para que, unido a las capas adicionales de análisis, se pueda crear un modelo mucho más robusto.

En la sección del análisis exploratorio se pudo extraer información importante de la data como un todo (toda su distribución), el cual a primera vista ayuda mucho al análisis preliminar de una data.

El preprocesamiento y selección de características son campos muy extensos que involucran mucho conocimiento de la lógica del negocio al que pertenece los datos, sin embargo, hay pasos que son muy generales en todo el ciclo de ciencia de datos, y la presente propuesta integra estos pasos de manera no supervisada hasta cierto punto. Estos pasos involucran el trabajo de valores NaN, escalamiento de variables, y selección de las mismas.

El módulo de evaluación (Evaluate) puede modelar de manera general problemas de clasificación y regresión, brindando una base de modelos con la cual empezar. Si bien es cierto la heurística para encontrar el mejor modelo tiene un carácter exploratorio (evaluación sobre un conjunto de modelos), los resultados indican que los puntajes obtenidos tienen una buena significancia tratándose de un proceso no supervisado.

Una exploración de modelos no era suficiente para estar seguros de haber escogido los mejores hiperparametros para la data de entrada. La búsqueda de hiperparámetros brindó

un camino para mejorar los puntajes de los modelos producto del módulo de evaluación, con esta técnica se pudo lograr aumentar en algunos puntos porcentuales la fiabilidad del modelo.

Por otro lado, en los experimentos se observó cosas interesantes. El experimento con el conjunto de datos Iris se logró obtener resultados prometeredores, esto debido básicamente a que la distribución de Iris es linealmente separable, esto es, que las fronteras que separan sus clases es una función lineal del tipo $y = wX$, donde y es la variable de respuesta y X es una matriz que engloba al conjunto de caracteristicas, esto permite a Pymach ajustarse bien a los datos con modelos simples, como se observa en la figura 6.6 . Adicionalmente, Iris sigue una ditribucion normal, provocando que funcione bien con modelos paramétricos (por ejemplo, *GaussianNB*, *LogisticRegression*, etc.) que esperan que la estructura de estos datos siga dicha distribución y en general, con un gran número de algoritmos, esta distribución normal se puede observar en la figura 6.1.

En el segundo experimento, data simulada de la av. Tupac Amaru, se puede concluir que el uso de Pymach y Sparkmach depende de la cantidad de datos, Pymach funciona bien con datos de mediana a baja dimensión, por el contrario, Sparkmach con datos más masivos, ver figura. Esto logra comprender de manera gráfica que es más costoso entrenar una data pequeña en Sparkmach que en Pymach, debido principalmente a la diferencia de paradigmas en las que estan basados cada uno de ellos. Pymach se basa en un paralelismo con memoria compartida y Sparmach en un paralelismo con memoria distribuida.

En el tercer experimento, datos de buses de Nueva York, se puede observar esta diferencia mucho más, el escalamiento que se logra al aumentar la data es directamente proporcional a la cantidad de nodos, como se observa en la figura 6.16. Este último gráfico confirma la idea de computación distribuida, que indica que para que un sistema basado en computación de alto rendimiento, como lo es un clúster de computadoras, funcione de manera óptima, se necesita tener una cantidad considerable de datos, para que la comunicación entre los nodos del sistema pueda ser menor en tiempos que la misma computación del problema.

Finalmente, en el último experimento, datos de una simulacion física, se puede concluir que al aumentar el tamaño de los datos Sparkmach puede obtener los mismos resultados en términos de predicción a un bajo costo computacional, esto se refleja en la disminución considerable de los tiempos de ejecución.

7.2. Trabajo futuro

En la sección del análisis exploratorio de la data (Analyze) los gráficos de salida son interpretables fácilmente si el usuario final tiene alguna noción de dichos ploteos estadísticos. Sin embargo, la filosofía de Pymach y Sparkmach es abstraer toda la complejidad técnica, para tal fin, en un trabajo futuro se necesitaría un módulo de interpretación de dichos gráficos usando textos explicativos autogenerados dependiendo de la distribución de la data, con el objetivo que al ver un gráfico este pueda autoexplicarse al igual que un ser humano lo entiende.

En el módulo de preparación (Prepare) hay técnicas de mapeo de variables que involucran tratar con variables categóricas (ordinales y cardinales) la cual mejoraría el modelo en problemas de clasificación y regresión.

En el apartado de selección de características (Selection) se puede aplicar de manera general la ingeniería de selección de características (feature engineering) el cual no solo selecciona las mejores características sino también la extración y modificación de las mismas.

En la seccion de evaluación de modelos (Evaluate) se puede aplicar la técnica de meta aprendizaje (meta-learning) que está siendo usada actualmente, por ejemplo, paquetes como AutoML y AutoSklearn lo están explotando mucho. El meta aprendizaje se inspira en el teorema "No free lunch"que indica que si una solución es buena para un tipo de problema, esta solución no tendría necesariamenteque que serlo para otros. Lo más recomendable seria explorar muchas técnicas y transferir dicho conocimiento a la resolución de problemas posteriores, básicamente tratar de ' aprender cómo aprender ' [43].

Finalmente, en el módulo de optimización de hiperparámetros (para el caso de pymach, Improve) se podría aplicar no solo la búsqueda de parámetros de manera exhaustiva sino también siguiendo una distribución, y enfocándose en los parámetros más probables (búsqueda aleatoria). Asimismo, técnicas de meta aprendizaje pueden ser aplicados en este contexto, ya que si un problema es resuelto con un conjunto de hiperpárametros, este conocimiento puede ser luego transferido hacia otro problema que comparta similitudes en sus características, logrando una reducción de tiempo en la búsqueda del conjunto de parámetros.

Por otra parte, hasta el momento solo se hace usado técnicas pertenecientes al Machine Learning, tales como, algoritmos de regresion, de arboles de decisiones, bayesianos, clustering, redes neuronales, etc. Sin embargo, atacar problemas con redes neuronales

profundas, *Deep Learning*, es un añadido que vale la pena explorar, ya que estos algoritmos aportan mucho en la parte de la ingeniería de características que en una primera instancia Pymach no lo realiza a profundidad.

Anexos A

Anexo

A.1. Manual de uso

A.1.1. Ingesta de datos

El primer paso para ejecutar Pymach es subir un archivo de texto. Este archivo de texto tiene que tener la variable de respuesta con el nombre de 'class', esta variable será reconocida por los demás módulos.

En la figura A.1 se hace click en el botón 'Choose File', se escoge el archivo a subir luego click en el botón 'Upload'. Este archivo aparecerá en el apartado de archivos subidos.

A.1.2. Análisis exploratorio

Luego de subir el archivo, el siguiente paso sería el de realizar un análisis exploratorio de los datos, esto se logra con el modulo 'Analyze', en la barra izquierda hacer click en 'Analyze' como se muestra en la figura A.2.

En la figura A.3 se muestra cierta metadata del archivo subido. Esta metadata es entrada para los siguientes módulos.

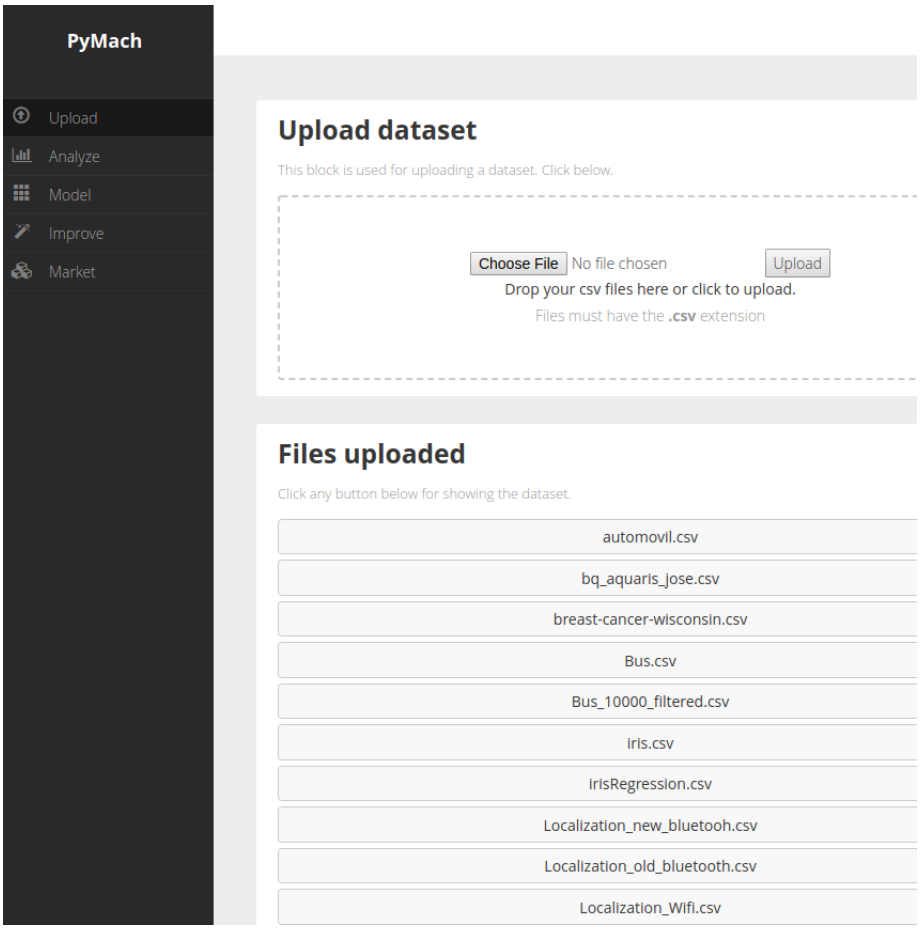


FIGURA A.1: Carga de un archivo csv a la plataforma.

Los gráficos estadísticos correspondientes a este conjunto de datos se observan en la figura A.4. Se observa el histograma, diagrama de cajas y matriz de correlación. Opcionalmente también se puede mostrar la matriz de dispersión.

A.1.3. Modelamiento

La parte más importante de Pymach se centra en el modelamiento en sí, en la figura A.5 se muestra el modulo 'Evaluate', el cual permite escoger el tipo de problema a resolver, sea de clasificación o de regresión y la data a modelar. Este módulo internamente se encarga de realizar la preparación (módulo 'Prepare'), selección (módulo 'Selection') y modelamiento (módulo 'Evaluate').

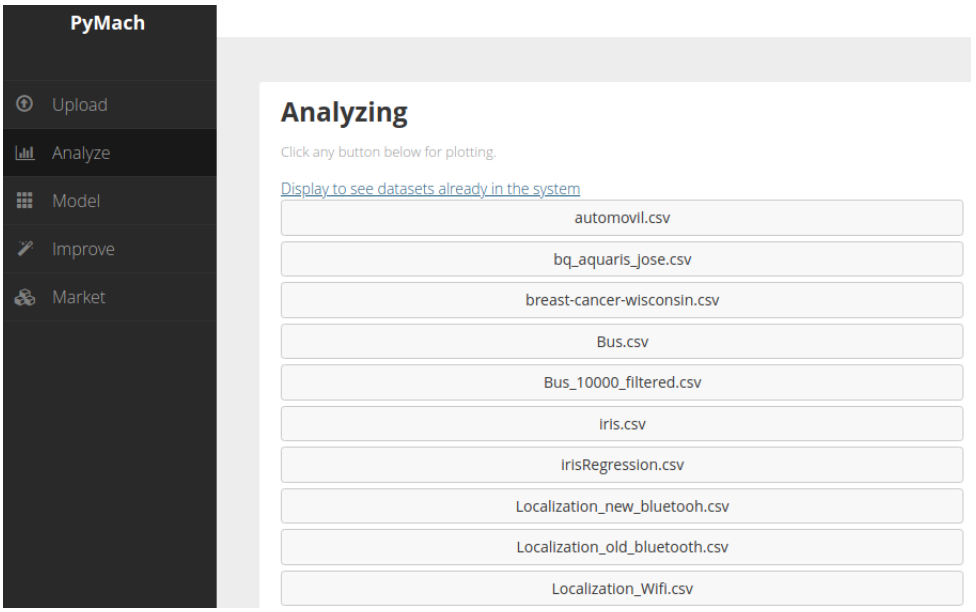


FIGURA A.2: Modulo Analyze. Ubicación del módulo en pymach.

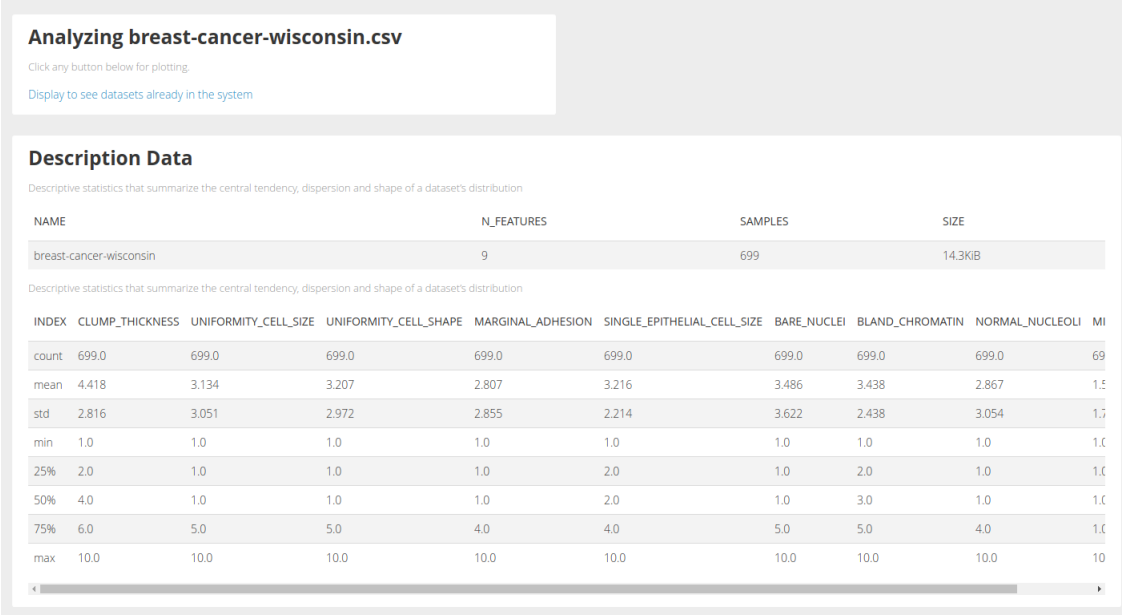


FIGURA A.3: Descripción de un conjunto de datos. Indica su nombre, cantidad de características, cantidad de filas, y tamaño.

A.1.4. Mejora del modelo

En la figura A.6 se muestra el modulo 'Improve', con este módulo se puede aumentar el puntaje de un módulo mediante la búsqueda exhaustiva de un conjunto de parámetros para un determinado modelo.

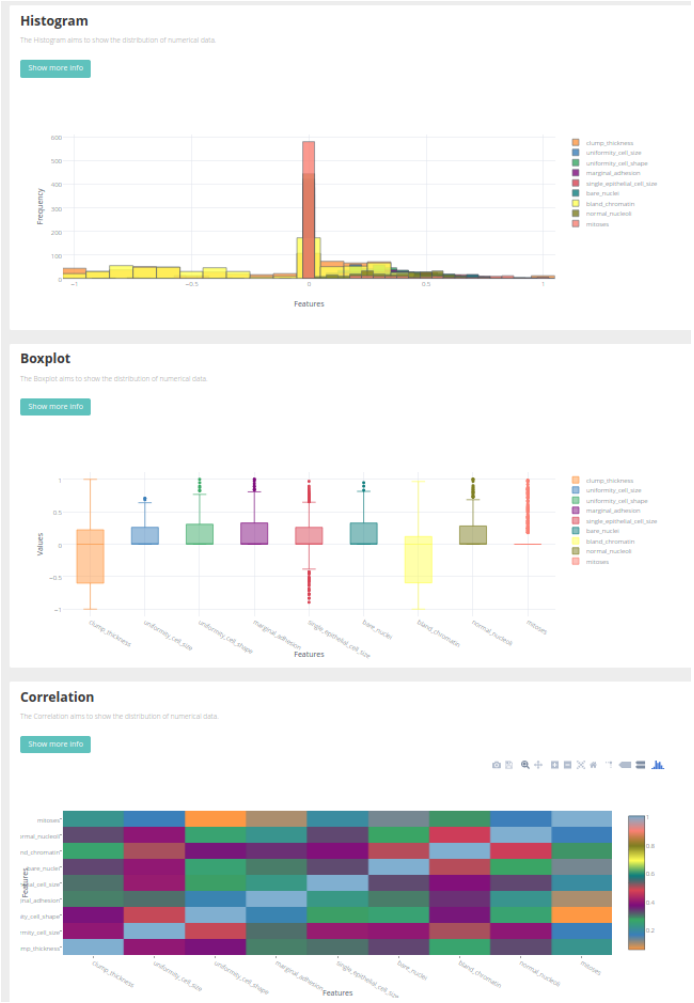


FIGURA A.4: Modulo Analyze. Se observan los gráficos que describen los datos.

A.1.5. Dashboard

Todos los resultados producto de los diferentes modulos se plasman en un dashboard final donde se ubica el análisis exploratorio y los resultados del modelamiento. Esto se observa en la figura A.7.

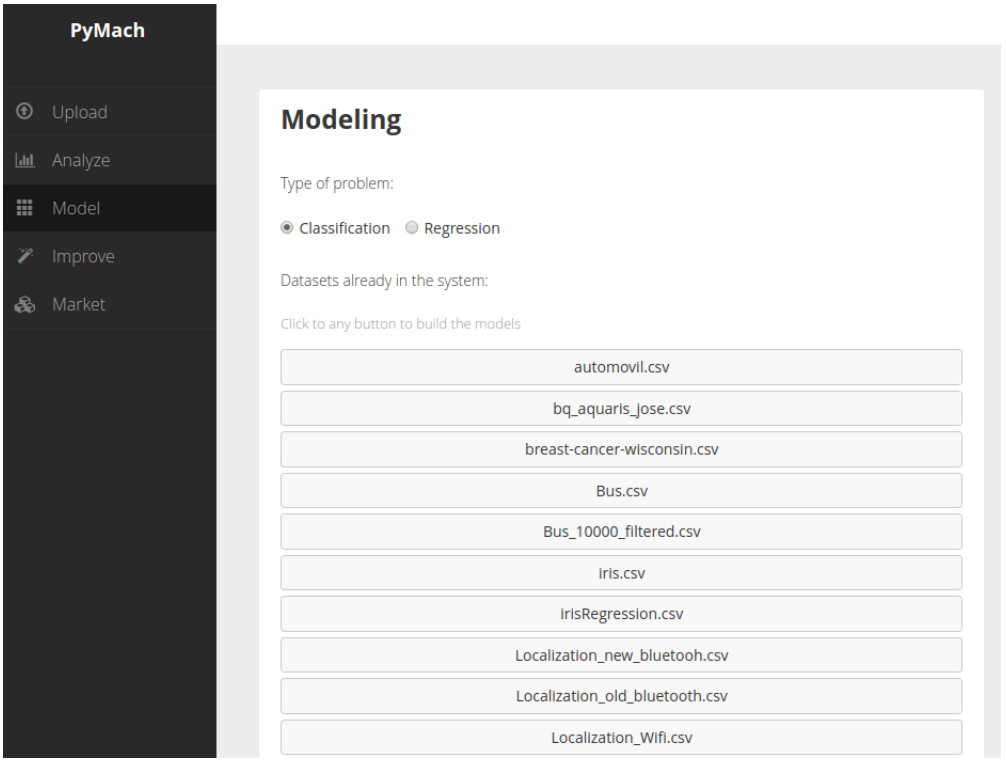


FIGURA A.5: Modulo de modelamiento. Al escoger el tipo de problema y la data, el módulo comienza a ejecutarse.

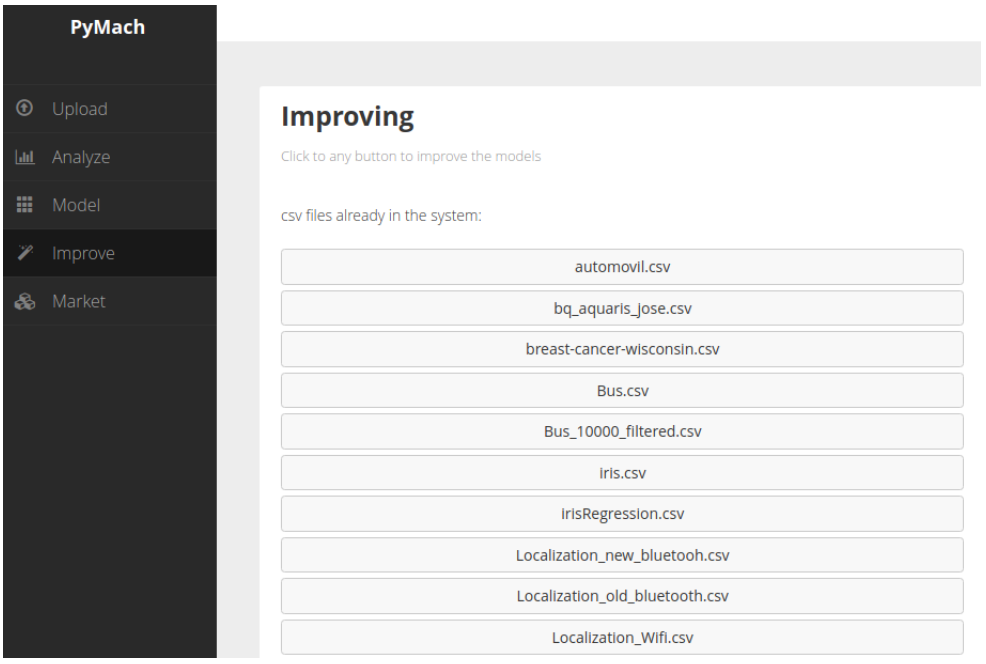


FIGURA A.6: Módulo Improve. Módulo que te permite hacer una búsqueda de los mejores hiperparámetros para un modelo.

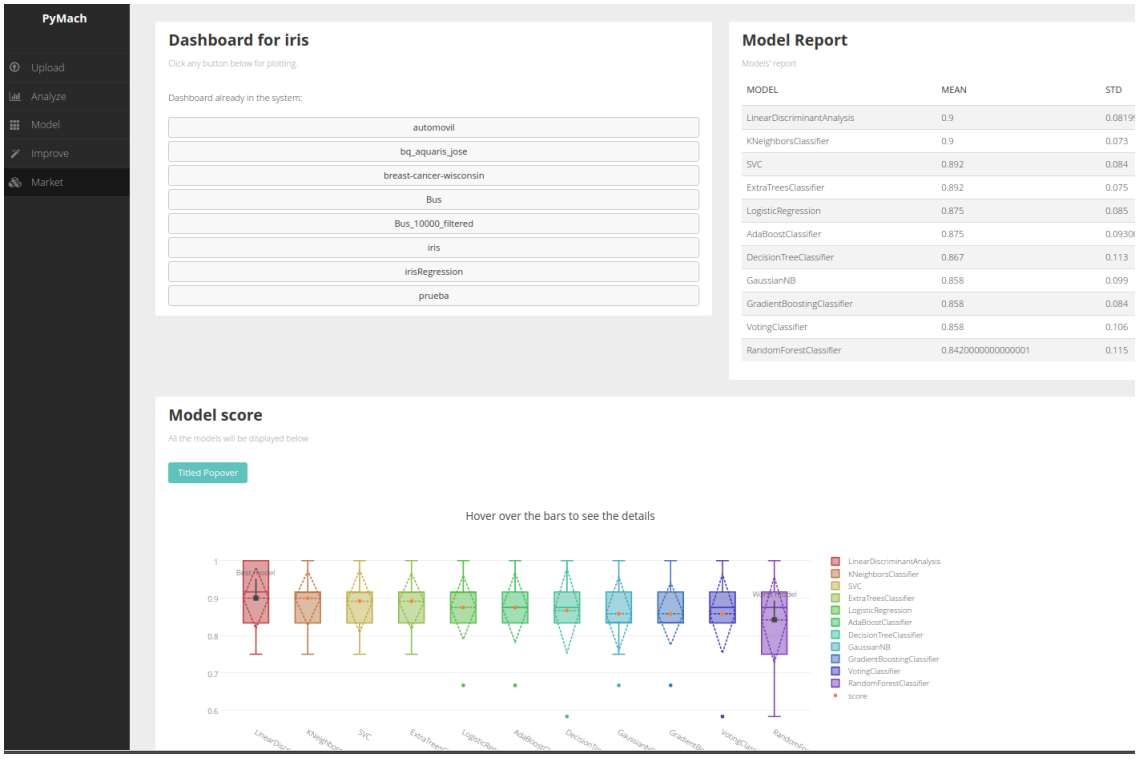


FIGURA A.7: Dashboard final. Muestra los resultados de todos los pasos.

Bibliografía

[1] Vince Jeffs. Automated intelligence map. Último acceso: 2018-05-07. URL vincejeffs.com.

[2] Elizabeth M. Roger D. Peng. The art of data science. *Leanpub*, 2015, pag 5-6.

[3] Katharina E. Matthias F., Aaron K. Efficient and robust automated machine learning. *University of Freiburg, Germany*, page 2–4, 2015.

[4] Scikit learn developers. Choosing the right estimator. Último acceso: 2018-05-07. URL http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html.

[5] Big Data Academy. Apache spark training mumbair. Último acceso: 2018-05-07. URL http://www.hadooptrainingchennai.in/apache-spark-training-mumbai/.

[6] Latent View. The apache hadoop stack. Último acceso: 2018-05-07. URL https://www.latentview.com/.

[7] Jason Brownlee. Master machine learning algorithms. 2016, pag 52-53.

[8] Vahid Mirjalili Sebastian Raschka. Python machine learning. *PACK*, 2017, pag. 255.

[9] Vahid Mirjalili Sebastian Raschka. Python machine learning. *PACK*, 2017, pag. 548-549.

[10] Trevor H. Robert T. Gareth J., Daniela W. An introduction to statistical learning. *Springer*, 2017, pag. 337-353.

[11] Trevor H. Robert T. Gareth J., Daniela W. An introduction to statistical learning. *Springer*, 2017, pag. 303-316.

[12] Yoshua Bengio James Bergstra. Random search for hyper-parameter optimization. *Journal of Machine Learning Research 13 (2012) 281-305*, 2012, pag. 303-316.

Último acceso: 2018-05-07. URL http://jmlr.csail.mit.edu/papers/volume13/bergstra12a/bergstra12a.pdf.

[13] Gusseppe Bravo-Rocca Piero Torres-Robatty, Jose Fiestas-Iquira. Sparkmach: A distributed data processing system based on automated machine learning for big data. *Springer Communications in Computer and Information Science (CCIS) Series*, 2018.

[14] Economics Network. Simple cellular automata on a spreadsheet. Último acceso: 2018-05-08. URL https://www.economicsnetwork.ac.uk/cheer/ch17/hand.htm.

[15] Andreas Schadschneider. Traffic flow modelling. Último acceso: 2018-05-08. URL http://www.thp.uni-koeln.de/~as/Mypage/traffic.html.

[16] Spark mooc. Spark tutorial. Último acceso: 2018-05-09. URL http://spark-mooc.github.io/web-assets/images/partitions.png.

[17] Andrew Ng. Coursera, machine learning. Último acceso: 2017-10-11. URL https://www.coursera.org/learn/machine-learning/.

[18] Birman K. y Vogels W. Renesse R. Astrolabe: A robust and scalable technology for distributed system monitoring, ma- nagement, and data mining. *ACM Transactions on Computer Systems*, (3):164–206, 2003.

[19] Gusseppe Bravo-Rocca Manuel Castillo-Cara. Citizen security using machine learning algorithms through open data. *8th IEEE Latin-American Conference on Communications LATINCOM*, 2016.

[20] Peter Norving Stuart J.Russel. Artificial intelligence, a modern approach. 2010, pag. 2.

[21] Tom M. Mitchell. Machine learning. *WCB McGraw-Hill*, page 2–4, 1997.

[22] Kevin P. Murphy. Machine learning: A probabilistic perspective. *University of British Columbia, Canada*, 2011, pag 1-2.

[23] Jeffrey Stanton. An introduction to data science. *Syracuse University, EEUU*, 2012, pag 2.

[24] Jason Brownlee. Machine learning mastery with python. 2016, pag 31-100.

[25] P. Christensson. Python definition. 2017. Último acceso: 2018-05-07. URL https://techterms.com.

[26] TIOBE. Programming community index. 2017. Último acceso: 2018-05-07. URL https://www.tiobe.com/tiobe-index/.

[27] Scikit-learn. Scikit-learn testimonials. 2017. Ultimo acceso: 2018-05-07. URL http://scikit-learn.org/stable/testimonials/testimonials.html.

[28] Jason Brownlee. Machine learning mastery with python. 2016.

[29] Vahid Mirjalili Sebastian Raschka. Python machine learning. *PACK*, 2017, pag. 55.

[30] Scikit-learn. Preprocessing and normalization. Último acceso: 2018-05-07. URL http://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing.

[31] Sebastian Raschka. Naive bayes and text classification i. *arXiv*, 2014, pag. 3. Último acceso: 2018-05-07. URL https://arxiv.org/pdf/1410.5329v3.pdf.

[32] Jason Brownlee. Machine learning mastery with python. 2016, pag 107-109.

[33] Patric Wendell Matei Zaharia Holden Karau, Andy Konwinski. Learning spark, lightning-fast data analysis. *O'REILLY*, 2015.

[34] Pyspark. Extracting, transforming and selecting features. Último acceso: 2018-05-07. URL https://spark.apache.org/docs/latest/ml-features.html.

[35] Pyspark. Classification and regression. Último acceso: 2018-05-07. URL https://spark.apache.org/docs/latest/ml-classification-regression.html#generalized-linear-regression.

[36] Pyspark. Classification and regression. Último acceso: 2018-05-07. URL https://spark.apache.org/docs/latest/ml-classification-regression.html#survival-regression.

[37] Cornel University. What is survival analysis? URL https://www.cscu.cornell.edu/news/statnews/stnews78.pdf.

[38] Öznur Yeldan. A stochastic continuous cellular automata traffic model with fuzzy decision rules. *POLITECNICO DI MILANO*, 2012.

[39] Qiuyan Huang Pengyang Xie Hui Jiang, Zhenpei Zhang. Research-of-vehicle-flow-based-on-cellular-automaton-in-different-safety. September 2015.

[40] Lukáš Sekanina Pavol Korček and Otto Fučík. A scalable cellular automata based microscopic traffic simulation. *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011.

[41] Long Island Rail Road Metro-North Metropolitan Transportation Authority. MTA — Subway, Bus. Metropolitan transportation authority. mta — subway, bus, long island rail road, metro-north. 2014. Last accessed 7 May 2018. URL http://web.mta.info/developers/MTA-Bus-Time-historical-data.html.

[42] Machine Learning Repository. Hepmass dataset. *UCI*, 2014, pag. 3. Last accessed 7 May 2018. URL https://archive.ics.uci.edu/ml/datasets/HEPMASS.

[43] Włodzisław Duch. Meta-learning. *Nicolaus Copernicus University, Poland.*