Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Matemáticas y Física

Maestría en Ciencia de Datos

MODELOS DE APRENDIZAJE AUTÓNOMO PARA CLASIFICACIÓN APLICADO A VARIACIONES EN EL VALOR DE ACCIONES

TESIS para obtener el **GRADO** DE
MAESTRO EN CIENCIA DE DATOS

Una tesis presentada por:
José Luis Ancira Pérez

Asesor de Tesis:
Dra. Rocío Carrasco Navarro

Tlaquepaque, Jalisco, Enero, 2023

Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Matemáticas y Física

Forma de Aprovación Maestría en Ciencia de Datos

Título de tesis: **MODELOS DE APRENDIZAJE AUTÓNOMO PARA CLASIFICACIÓN APLICADO A VARIACIONES EN EL VALOR DE ACCIONES**

Autor: **José Luis Ancira Pérez**

Tesis Aprobada para completar todos los requisitos de grado para la Maestría en Ciencia de Datos.

Asesor de Tesis, **Dra. Rocío Carrasco Navarro**

Revisor de Tesis, **Dr. Fernando Ignacio Becerra López**

Revisor de Tesis, **Dra. Diana Paola Montoya Escobar**

Asesor Académico, **Dra. Rocío Carrasco Navarro**

Tlaquepaque, Jalisco, Enero, 2023

MODELOS DE APRENDIZAJE AUTÓNOMO PARA CLASIFICACIÓN APLICADO A VARIACIONES EN EL VALOR DE ACCIONES

José Luis Ancira Pérez

Resumen

La clasificación de acciónes del mercado de valores, entre aquellas que puedan incrementar o no su valor año contra año, es uno de los temas con mayor complejidad en el mundo del análisis de datos debido a la gran cantidad de variables que afectan su resultado como lo son el desempeño/resultado de las compañías, cambios organizacionales, variables económicas externas, percepción del mercado así como otras dinámicas intrínsecas.

Este trabajo se centrará en un modelo de aprendizaje autónomo o machine learning, en inglés, para clasificación de acciones basados en los reportes y resultados financieros trimestrales de algunas compañías públicas de Estados Unidos de América, con el fin de deteriminar o categorizar aquellas acciones que pueden generar mayor valor y aquellas que no.

Los resultados contribuyen a la discusion sobre si es posible aplicar satisfactoriamente modelos de "machine learning", así como extraer resultados satisfactorios, a las dinámicas de los mercados de valores, sus activos y movimientos en los precios, partiendo únicamente con la información de los resultados financieros de las compañías.

Índice general

Índice de figuras

Page

Índice de tablas

Page

A mis padres

1 Introducción

Contents

1.1 Justificación

Cada trimestre las compañías públicas, aquellas que han emitido valores para financiarse, están obligadas a publicar sus resultados financieros del período. En Estados Unidos, las reglas de la Comisión Nacional del Mercado de Valores (SEC por sus siglas en inglés)[1] requieren que las empresas presenten informes anuales e informes trimestrales de forma continua[2]. En dichos informes de resultados se reporta información del estado de resultados (ingresos, costos, gastos, etc.), del balance general (activos, pasivos y capital) e incluso del flujo de efectivo. La razón de estos requisitos de información es mantener informados a los accionistas y a los mercados de forma periódica y transparente. Los informes presentados ante la SEC pueden ser vistos por el público en el sistema de obtencion, análisis y recuperación de datos (SEC EDGAR por sus siglas en ingles) [3].

Los valores u acciones emitidas por las compañías van cambiando de valor/precio a lo largo del tiempo dependiendo de la oferta y demanda por esas acciones. Pueden existir grandes ganancias o pérdidas para los accionistas dependiendo del precio en el que compran y venden dichas acciones. Por esta razón, existe mucho interés en encontrar métodos y modelos que puedan anticiparse a los precios y rendimientos de estas acciones así como, de poder clasificar las compañías que puedan incrementar los rendimientos de sus acciones y aquellas que no.

El mercado de valores (conjunto de bolsas de valores) es el espacio

[1] sec.gov. La SEC: Lo que Somos, y lo que Hacemos. https://www.sec.gov/investor/espanol/quehacemos.htm, September 2001

[2] PLLC . Anthony L.G. Public Company SEC Reporting Requirements . http://www.legalandcompliance.com/securities-resources/sec-requirements-for-public-companies/

[3] U.S. Securities and Exchange Commission. https://www.sec.gov/edgar.shtml, January 2017

donde las compañías públicas acuden para obtener financiamiento, captando el ahorro de los inversionistas, a quienes les ofrecen distintos beneficios y rentabilidad en función del título de la compañía que adquieran.

El comercio de acciones era principalmente una actividad basada en certificados físicos y existía la necesidad de profesionales que estuvieran físicamente en el área de compra/venta para realizar movimientos de acciones. Hoy, estos certificados han sido reemplazados por su forma electrónica y pueden registrarse y transferirse electrónicamente.

Estas transacciones electrónicas han permitido el ingreso de algoritmos para comprar o vender acciones de forma automática cuando se cumplan ciertas condiciones, esto es conocido como Algorithmic Trading. Actualmente, más del 80 por ciento de las transacciones de acciones se realizan a través de dichos algoritmos[4]. Se espera que el mercado de Algorithmic Trading aumente un 5.98 por ciento entre 2020 y 2025, lo que equivale a 3.79 miles de millones de USD.

[4] Yun Li. 80 percent of the stock market is now on autopilot . https://www.cnbc.com/2019/06/28/80percent-of-the-stock-market-is-now-on-autopilot.html, June 2019

Con esta digitalización se ha incrementado el interés en generar modelos que se anticipen a los movimientos de los mercados financieros y de modelos que clasifiquen a las empresas con mayor potencial de incrementar su valor y aquellas que no. La gran mayoría de los modelos matemáticos de finanzas cuantitativas se basan en el supuesto de que los precios del mercado evolucionan en el tiempo de acuerdo con un proceso estocástico (son aleatorios).

Existen distintos formas o métodos para predicción y clasificación del valor de acciones: [5]

[5] AMG Funds LLC. Fundamental vs. Technical Analysis. https://tinyurl.com/7bmb8zk

1. Análisis técnico. Determinan patrones del precio de las acciones a través del tiempo.

2. Análisis sentimental. El valor de las acciones también es determinado por el sentir del público y su actitud hacia las compañías. Si hay un sentimiento general negativo, el valor caerá y si hay un sentimiento positivo el valor tiende a subir.

3. Análisis fundamental. El cual se basa en analizar los resultados financieros de las compañías, así como factores económicos.

1.2 *Estado del Arte*

En los últimos años, se han realizado avances significativos en lineas de investigación de machine learning para predecir, clasificar y comprender las variables que influyen en la volatilidad de las acciones, sin embargo, hay mucho camino que recorrer ya que aún no hay modelos que funcionen de forma consistente o sostenida. [6]

[6] Abhishek Gupta. A Survey on Stock Market Prediction Using Various Algorithms . https://www.academia.edu/43646075/A_Survey_on_Stock_Market_Prediction_Using_Various_Algorithms, March 2014

Se habla en la sección anterior de como el precio de la acción se modifica de acuerdo a resultados de las compañías así como del sentir y las percepciones del público que a la vez afectan su demanda y de como se aplican análisis técnicos, sentimentales y fundamentales en el aprendizaje autónomo para predecir y clasificar resultados de acciones. Se suelen usar resultados financieros en el análisis fundamental, tendencias de los precios en el análisis técnicos e información de noticieros y redes sociales (principalmente tweets publicos de la red social twitter) en el análisis sentimental.

Aún con el conjunto de información con los que se cuenta, se han creado modelos de aprendizaje autónomo con resultados de mediana precisión.[7].

En los últimos 10 años, la industria financiera ha invertido muchos recursos para aplicar modelos complejos para predicción en el mercado de valores. Se ha encontrado una proporción muy alta de modelos de predicción con una baja cantidad de modelos de clasificación.

Aún con estas inversiones para crear modelos en el mercado de valores, los resultados siguen generando dudas sobre si un modelo complejo es suficiente para predecir y clasificar satisfactoriamiente a los mercados financieros. Shivam Sinha menciona que "la inteligencia artificial no es aún un sustituto para la inteligencia humana al menos en los mercados financieros, hablando que es importante elegir variables de entrada basados en inteligencia humana e intuición económica y no solo de información financiera e historicos debido a que la información financiera contiene información irrelevante, es decir, la información finaicera contiene una baja relación señal/ruido" [8].

A pesar de las cuestiones mencionadas que impactan negativamente a los modelos de aprendizaje autónomo aplicadas en el mercado de valores, sí existen algunos modelos que arrojan resultados moderadamente satisfactorios como el de Yang Bai donde se aborda la asignación de portafolios de inversión como un problema de clasificación para selección de acciones [9].

Muchos otros modelos, como el de Fernando Villada, Nicolás Muñoz y Edwin García [10] se centran en predicción del precio de uno o pocas acciones especificamente seleccionadas, dejando un espacio para modelos que estudien una amplia cantidad de acciones.

1.3 *Objetivos*

Como se mencionó en la sección anterior, es importante desarrollar modelos de clasificación aplicados a los mercados de valores debido a la gran cantidad de modelos enfocados exclusivamente en predicción, dejando un área de oportunidad para la clasificación.

[7] Jin Liu Sohrab Mokhtari, Kang K. Yen. Effectiveness of Artificial Intelligence in Stock Market Prediction based on Machine Learning. https://arxiv.org/abs/2107.01031, June 2021

[8] Shivam Sinha. Making Machine Learning Work For Financial Market Prediction. https://www.forbes.com/sites/forbesfinancecouncil/2021/10/18/making-machine-learning-work-for-financial-market-?sh=1c5e5a8e2cce, October 2021

[9] Yang Bai. Machine Learning Classification Methods and Portfolio Allocation: An Examination of Market Efficiency. https://herbert.miami.edu/_assets/pdfs/faculty-research/business-conferences/machine-learning/yang-bai.pdf, January 2021

[10] Nicolás Muñoz y Edwin García Fernando Villada. Aplicación de las Redes Neuronales al Pronóstico de Precios en el Mercado de Valores. https://scielo.conicyt.cl/scielo.php?pid=S0718-07642012000400003&script=sci_arttext&tlng=en, November 2012

1.3.1 Objetivo General

El objetivo general de este trabajo es utilizar modelos de aprendizaje auntomático para clasificar el desempeño de una acción a partir de la información financiera publicada para la SEC y sus indicadores financieros.

1.3.2 Objetivos Específicos

Implementar tres diferentes modelos de clasificación (regresión logística, arbol de decisión y red neuronal) para clasificar las diferentes acciones entre aquellas que generan valor año contra año y comparar los resultados arrojados de cada modelo para así identificar el mejor modelo para este caso.

1.4 Organización de la tesis

En el capitulo 2 se muestra una descripción de la base de datos donde se explica los tipos de variables, se especifica la cantidad de datos y valores faltantes. A su vez, se describen las herramientas y mecanismos utilizados para la imputación de los datos.

El capítulo 3 se habla del marco teórico y los fundamentos matemáticos de los modelos utilizados.

En el capitulo 4 se aborda el desarrollo metodológico y se describen, discuten y comparan los resultados del presente trabajo.

Por último, las conclusiones y el trabajo futuro de esta tesis se presentan en el capítulo 5.

El presente trabajo de ciencia de datos siguió un flujo de trabajo representado en el diagrama de la figura 1.1.

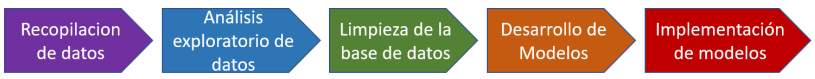


Figura 1.1: Diagrama de flujo

2 Análisis y preparación de los datos

Contents

Para este trabajo se utilizaron dos bases de datos obtenidas de diferentes fuentes.

La primera base de datos se obtuvo de la página de internet de Kaggle [1] que a su vez fue extraida de Financial Modeling Prep API [2]. La segunda base de datos se obtuvo de FactSet. solución de software que provee acceso a información financiera y analítica con la cual inversionistas pueden hacer toma de decisiones.

A continuación se hará una descripción detallada de cada una de las bases de datos.

[1] Kaggle. 200+ Financial Indicators of US stocks (2014-2018)). https://www.kaggle.com/cnic92/200-financial-indicators-of-us-stocks-20142018

[2] Financial Modelling Prep. Financial Modelling Prep. https://site.financialmodelingprep.com/developer/docs

2.1 Base de Datos (dbA)

La primer base de datos (dbA) cuenta con 224 indicadores financieros que datan del 2014 al 2018 con información de más de 4 mil empresas (acciones) distintas. Estos indicadores financieros contienen información de los tres estados financieros (estado de resultados, Balance general, flujo de efectivo).

La tabla 2.1 muestra las primeras 35 variables de la base de datos dbA. Esta base de datos contiene información del estado en el que se encuentran las compañías financieramente. A través de ellas se puede analizar su salud contable, financiera, de flujo de efectivo, retornos de inversión e incluso hay parámetros para comparar cada compañía con los resultados promedio de su industria conrrespondiente.

La base de datos contiene una variable dependiente llamada ”Class” o ”Clase” la cual es binaria. Los ceros indican una disminución en el precio de la acción de determinada compañía comparada con su precio del año anterior, mientras que los unos son para aquellas compañias

que aumentaron el precio de su acción año contra año.

2.1.1 *Limpieza de la base de datos dbA*

Uno de los principales problemas que se identificaron en la primer base de datos (dbA) fue que existía un gran número de valores faltantes. El primer paso para la limpieza de la base de datos fue eliminar aquellas variables con mayor cantidad de valores faltantes y, los valores faltantes restantes, completarlos con la media del sector a la que corresponde cada compañia de cada variable.

Una vez realizada esta limpieza y eliminación de variables pasamos de una base de datos de 4392 compañias y 224 indicadores (4392X224) a una de 1502 compañías y 216 indicadores (1502X216). El criterio para reducir las filas y columnas que mas valores faltantes contenían fue: mantener aquellas variables (columnas) que tuvieran al menos 900 valores non-NaN y eliminar las que tuvieran menos. Mantener aquellas compañias (filas) que tuvieran al menos 190 valores non-NaN y eliminar las que tuvieran menos de estos.

Se decidió este criterio para mantener aquellas variables con el 60 por ciento o más valores non-NaN, como lo sugiere Alvira Swalin.[3].

En la figura 2.1 vemos un histograma que muestra la distribución de la variable '2019 Price Var'. Tambien se muestra en dicha figura, una visualización de los valores faltantes por variable agrupados de mayor a menor cantidad (barras de color azul) para la base de datos dbA y el cumulativo de valores faltantes representado por la línea roja.

[3] Alvira Swalin. How to Handle Missing Data. https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4, January 2018

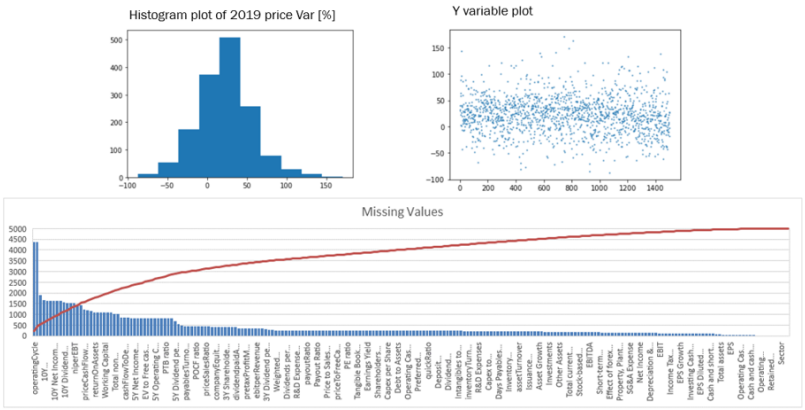


Figura 2.1: Histograma y grafico de valores faltantes dbA

Por último, en esta sección de limpieza de la base de datos, se escalaron los datos para de esta forma transformarlos todos dentro de un mismo rango utilizando MinMaxScaler. Debido a que los algoritmos de clasificación funcionan de manera más óptima con datos escalados

al calcular distancias entre puntos.

Este estimador escala y traduce cada característica individualmente de manera que se encuentre en el rango dado en el conjunto de entrenamiento, por ejemplo, entre cero y uno.[4]. Se eligió MinMaxScaler debido a que mantiene la forma de la distribución de la base de datos, mientras que otros, como el Standard Scaler, transformarían la distribución a media cero y desviación estandar de 1.

[4] Runebook.dev. `sklearn.preprocessing.MinMaxScaler`. https://runebook.dev/es/docs/scikit_learn/modules/generated/sklearn.preprocessing.minmaxscaler

2.2 *Base de Datos dbB*

La segunda base de datos (dbB) cuenta con menor cantidad de información, 200 empresas (acciones) sin embargo, es información más reciente con resultados del año 2020. Contiene 34 variables o indicadores financieros extraídos de la base de datos de FactSet Research Systems Inc.

La tabla 2.2 muestra todas las variables de la base de datos dbB. Esta base de datos contiene información del estado en el que se encuentran las compañías financieramente. A través de ellas se puede analizar su salud contable, financiera, de flujo de efectivo y retornos de inversión. Al igual que la base de datos dbA, esta base de datos contiene una variable dependiente llamada "Class" o "Clase".

Para iniciar con la exploración de los datos se requiere visualizar nuestra metadata. La siguiente gráfica de barras nos muestra una forma simple de visualizar rápidamente porcentaje de variables discretas, continuas, valores u observaciones faltantes, etc. Lo cual nos da una guía al momento de limpiar la base de datos.

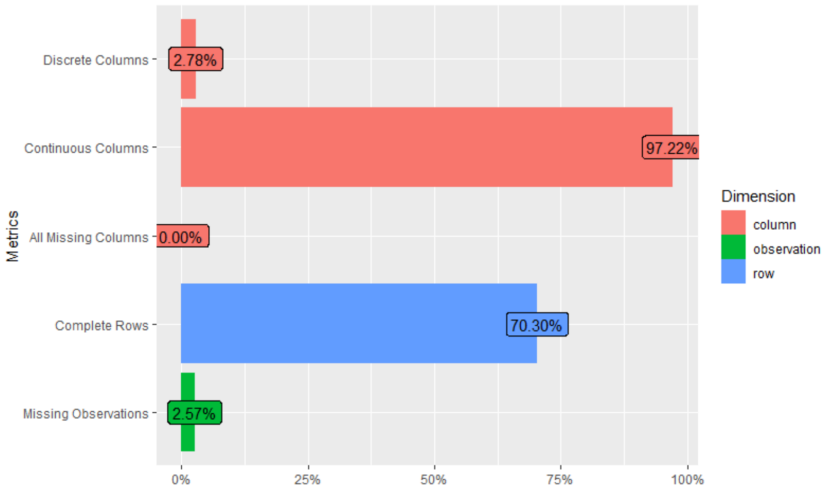


Figura 2.2: Intro: Base de Datos dbB

La figura 2.3 muestra las estadisticas descriptivas de la segunda base de datos de FactSet (dbB), donde podemos observar las medidas de

tendencia central y medidas de dispersión de las primeras 10 variables.

	Company	Price Change (%)	+/ S&P 5 (%)	+/ Industry (%)	Div Yld (%)	Sales	EBITDA	EBIT	Net Inc	EPS (Dil)	Divs PS
count	200	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000
unique	194	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	HHC	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	53.618389	53.627495	48.927275	1.701731	12024.625650	8514.425291	8527.142299	8247.606133	13.498583	6.699370
std	NaN	100.384669	96.486313	90.211026	4.688344	23947.710539	29911.216606	31069.195249	29848.783356	123.223745	84.815655
min	NaN	0.400000	0.271194	0.020000	0.000000	0.000000	0.784000	1.475000	0.456000	0.000000	0.000000
25%	NaN	12.504050	13.246625	11.128175	0.000000	491.497000	90.182750	62.111250	54.795000	0.651800	0.000000
50%	NaN	24.309000	26.470350	23.203450	0.000000	1856.210000	436.474500	281.443500	233.544000	2.053000	0.000000
75%	NaN	43.660000	49.218125	42.684500	1.917466	10230.925000	2153.800000	1537.702500	1272.317500	3.967973	1.048125
max	NaN	743.437000	727.178000	643.612000	50.040000	135082.000000	292852.000000	301801.000000	268905.000000	1738.060000	1200.000000

Figura 2.3: Estadísticos de la base de datos dbB

Es importante conocer si la base de datos cuenta con una distribución balanceada de las clases, es decir, si la variable de salida tiene un cantidad similar de valores positivos que de valores negativos. Cuando una base de datos se encuentra desbalanceada, puede arrojar resultados erroneos. La mayoria de los data sets contienen cierto porcentaje de desbalance, no debería existir un impacto significatigo en el desempeño del modelo cuando estos desbalances son pequeños. Revisando la variable clase de la base de datos dbB obtenemos los siguientes resultados mostrados en la figura 2.4, donde observamos que hay un pequeño desbalance que no es significativo para el modelo y el cual se podrá evaluar mas adelante con la matriz de confusión.

Class	Freq	%
0	86	43%
1	114	57%

Figura 2.4: Distribución variable Clase

Adicionalmente, para identificar cualquier correlación existente entre las variables, se realizó una matriz de correlación, la cual mide el grado de relación lineal entre cualquier par de variables. Los valores de correlación caen entre -1 y +1, si dos variables tienden a aumentar o decrecer juntas, el valor de correlación es positivo y tenderá a +1.

En la figura 2.5 observamos que no existen dos variables con alta correlación entre ellas.

2.2.1 *Limpieza de la base de datos dbB*

Se observaba en la figura 2.2 un 2.57 porciento de valores faltantes, ahora se revisa con la siguiente gráfica de barras los valores faltantes de manera mas profunda. Veremos donde especificamente hace falta la información, al agruparla por columnas y encuentrar todas las filas faltantes por columna.

Figura 2.5: Matriz de Correlación

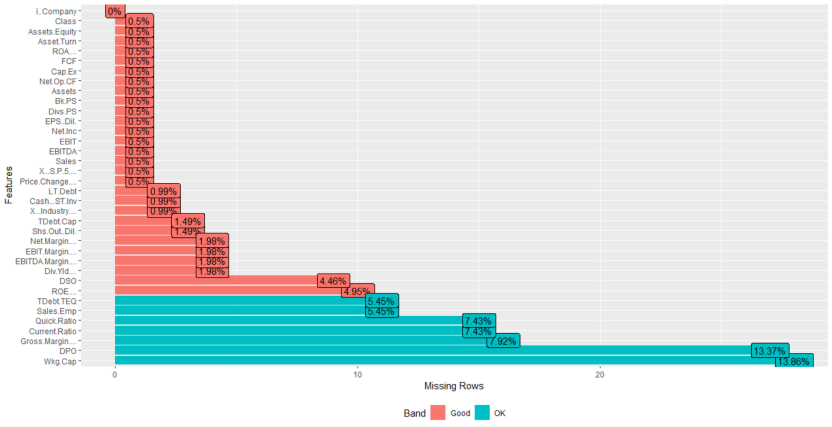


Figura 2.6: Valores faltantes dbB

Debido a que cuenta con pocos valores faltantes, no fue necesaria la eliminación de variables. Para esta base de datos se completaron los valores faltantes con valores aleatorios utilizando la función random de la libreria numpy. El valor aleatorio se determinó en un rango entre el valor mínimo y el máximo para cada variable. Este método para completar los valores se eligió en vez de media, moda o mediana ya que sabemos que los datos de las variables de esta base de datos dependen de los resultados fincieros de cada compañía y no necesariamente están relacionadas entré sí.

Al igual que para la primer base de datos, en ésta se escalaron los datos después de haber abordado el tema de los valores faltantes utilizando MinMaxScaler.

Tabla 2.1: Variables de la dbA

VARIABLE	DESCRIPCIÓN	UNIDAD
Revenue	Ingresos/Ventas	USD
Revenue Growth	Crecimiento sobre ventas	Porcentaje
Cost of Revenue	Costos	USD
Gross Profit	Margen bruto	USD
RD Expenses	Gastos de investigación y desarrollo	USD
SGA Expenses	Gastos de venta y administrativos	USD
Operating Expenses	Gastos operativos	USD
Operating Income	Ingresos operacionales	USD
Interest Expense	Gastos de intereses	USD
Earnings before Tax	Ganancias antes de impuestos	USD
Income Tax Expense	Impuesto sobre la renta	USD
Net Income - Non-Controlling int	Ingreso neto - interes	USD
Net Income - Discontinued ops	Ingreso neto - operaciones descontinuadas	USD
Net Income	Ingreso neto	USD
Preferred Dividends	Dividendos prefernciales	USD
Net Income Com	Ingreso neto para accionistas comunes	USD
EPS	Ganancias por acción	USD
EPS Diluted	Ganancias por acción diluidas	USD
Weighted Average Shs Out	Promedio ponderado de acciones en circulación	USD
Weighted Average Shs Out (Dil)	Promedio ponderado de acciones en circulación diluidas	USD
Dividend per Share	Dividendos por acción	USD
Gross Margin	Margen Neto	Porcentaje
EBITDA Margin	Margen EBITDA	Porcentaje
EBIT Margin	Margen EBIT	Porcentaje
Profit Margin	Margen de beneficio	Porcentaje
Free Cash Flow margin	Margen de flujo de efectivo	Porcentaje
EBITDA	Ingresos antes de impuestos, intereses, depreciación y amortización	USD
EBIT	Ingresos antes de impuestos e intereses	USD
Consolidated Income	Ingresos consolidados	USD
Earnings Before Tax Margin	Ingresos antes de margen de impuestos	USD
Net Profit Margin	Margen neto de ingresos	Porcentaje
Cash and cash equivalents	Efectivo y equivalencias	USD
Short-term investments	Inversiones a corto plazo	USD
Cash and short-term investments	Efectivo e inversiones a corto plazo	USD
Receivables	Cuentas por cobrar	USD

Tabla 2.2: Variables de la dbB

VARIABLE	DESCRIPCIÓN	UNIDAD
Price Change	Cambio Porcential del precio de la acción	Porcentaje
+/- SP 500	Comparación vs otras compañías del SP 500	Porcentaje
'+/- Industry	Comparación vs otras compañías de su industria	Porcentaje
Div Yld	Rentabilidad por dividendo	Porcentaje
Sales	Ventas/Ingresos	USD
EBITDA	Ingresos antes de impuestos, intereses, depreciación y amortización	USD
EBIT	Ingresos antes de impuestos e intereses	USD
Net Inc	Ingresos Netos	USD
EPS (Dil)	Ganancias por acción	USD
Divs PS	Dividendos por acción	USD
Shs Out (Dil)	Shs Out (Dil)	USD
Bk Ps	Bk por acción	USD
Cash ST Inv	Efectivo e inversiones a corto plazo	USD
Assets	Activos	USD
Wkg Cap	Capital de trabajo	USD
LT Debt	Deuda a largo plazo	USD
Net Op CF	Flujo de efectivo	USD
Cap Ex	Gasto de capital	USD
FCF	Flujo libre de caja	USD
Gross Margin	Margen bruto	USD
EBITDA Margin	Margen EBITDA	Porcentaje
EBIT Margin	Margen EBIT	Porcentaje
Net Margin	Margen neto	Porcentaje
ROA	Retorno sobre activos	Porcentaje
ROE	Retorno sobre acciones	Porcentaje
Asset Turn	Rotación de activos	USD
Assets/Equity	Activos /Capital	USD
Sales/Emp	Ventas/Emp	USD
DSO	Dias de ventas pendeintes	USD
DPO	Dias por pagar pendientes	USD
Current Ratio	Razón actual	USD
Quick Ratio	Razón rápida	USD
TDebt Cap	TDebt Cap	Porcentaje
TDebt TEQ	TDebt TEQ	Porcentaje
Class	Clase (1 o 0)	Clasificador

3 Fundamentos Matemáticos

Contents

El objetivo del presente trabajo busca clasificar si una compañía podrá aumentar el valor de su acción, al ser esta una variable cualitativa, se utilizarán modelos de clasificación.

En este capítulo se presentan los fundamentos matematicos de los modelos de redes neuronales, árboles de decisión y regresión logística.

3.1 Modelos de Clasificación

El aprendizaje automático u autónomo se centra en desarrollar sistemas que aprenden o mejoran el rendimiento en función de los datos que consumen. La clasificación es un tipo de aprendizaje autónomo que se usa para categorizar elementos en clases[1].

Dicho de otra manera, la clasificación es el proceso de reconocer, identificar y agrupar objetos e ideas en categorias predeterminadas. Los algoritmos de clasificación utilizados en el aprendizaje automático consumen datos de entrenamiento de entrada con el fin de predecir la probabilidad de que los datos que siguen se incluyan en una de las categorías predeterminadas. [2].

Existen dos métodos para aprendizaje automático: supervisado y no supervisado. En un modelo supervisado, se introduce en el algoritmo de clasificación un conjunto de datos de entrenamiento previamente etiquetados, es decir, que conocemos el valor de su atributo objetivo o clasificador. Los modelos no supervisados, por otro lado, se alimentan de un conjunto de datos que no están etiquetados y busca similitudes o patrones en los datos. La clasificación puede pertenecer a la categoría de aprendizaje supervisado cuando los objetivos o clases también se

[1] Trevor Hastie Rob Tishirani Gareth James, Daniela Witten. An Introduction to Statistical Learning with Applications in R. https://hastie.su.domains/ISLR2/ISLRv2_website.pdf, 2021

[2] Simplilearn. Everything You Need to Know About Classification in Machine Learning. https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning, November 2021

proporcionan junto con los datos de entrada. Ejemplos de este tipo de algoritmos son:

1. Redes neuronales: conjunto de unidades/neuronas artificiales, conectadas entre sí para transmitirse señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo unos valores de salida (predicciones o clasificaciones).

2. Arboles de decisión: toma una serie de decisiones en forma de árbol. En estas estructuras de árbol, las hojas representan etiquetas de clase y las ramas representan las conjunciones de características que conducen a esas etiquetas de clase.

3. Regresión logística: tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica en función de las variables independientes o predictoras. Es útil para modelar la probabilidad de un evento ocurriendo en función de otros factores

3.2 *Redes Neuronales*

Una red neuronal es un modelo simplificado que emula el modo en que el cerebro humano procesa la información. La red aprende examinando los registros individuales, generando una predicción para cada registro y realizando ajustes a las ponderaciones cuando realiza una predicción incorrecta. Este proceso se repite varias veces y la red sigue mejorando sus predicciones hasta haber alcanzado uno o varios criterios de parada [3].

Las redes neuronales permiten obtener un modelo no explícito que relaciona un conjunto de variables de salida con un conjunto de variables de entrada. Así, estos modelos permiten predecir cuál es el valor de salida, dados unos valores de entrada del modelo. Para estimar el modelo es necesario disponer de un conjunto de observaciones de las variables. Estas observaciones son usadas como patrones de entrenamiento para que la red aprenda y sea capaz de predecir una salida del modelo, ante nuevas observaciones[4]. En la figura 3.1 podemos observar un ejemplo de red neuronal, en este caso un perceptrón simple, con una sola capa de neuronas con una única salida.

Tambien existen las redes neuronales multicapa, las cuales se organizan en capas divididas en tres secciones: una capa de entrada, con unidades que representan los campos de entrada; una o varias capas ocultas; y una capa de salida. Las unidades se conectan con ponderaciones o pesos. Los datos de entrada se presentan en la primera capa, y los valores se propagan desde cada neurona hasta cada neurona de la capa siguiente. al final, se envía un resultado desde la capa de

[3] IBM Corporation. El modelo de redes neuronales. https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=networks-neural-model

[4] Jaime Ariel-Toral Barrera. Redes Neuronales. http://www.cucei.udg.mx/sites/default/files/pdf/toral_barrera_jamie_areli.pdf

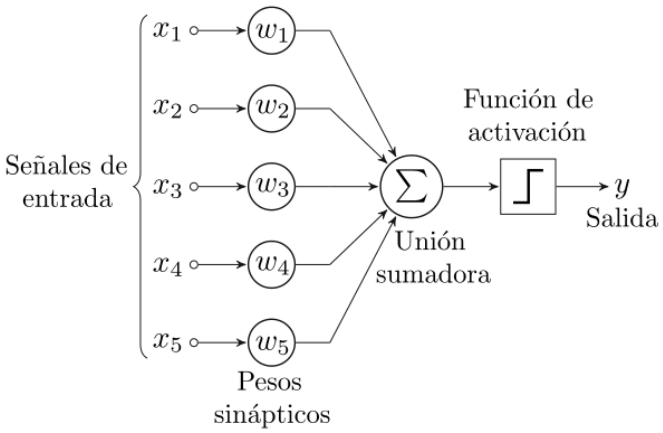


Figura 3.1: Modelo de perceptron simple

salida. En la figura 3.2 se puede observar la dinámica de las capas de una red neuronal multicapa, cabe resaltar que en este ejemplo la capa de salida muestra una sola neurona, sin embargo, puede haber varias neuronas en la capa de salida.

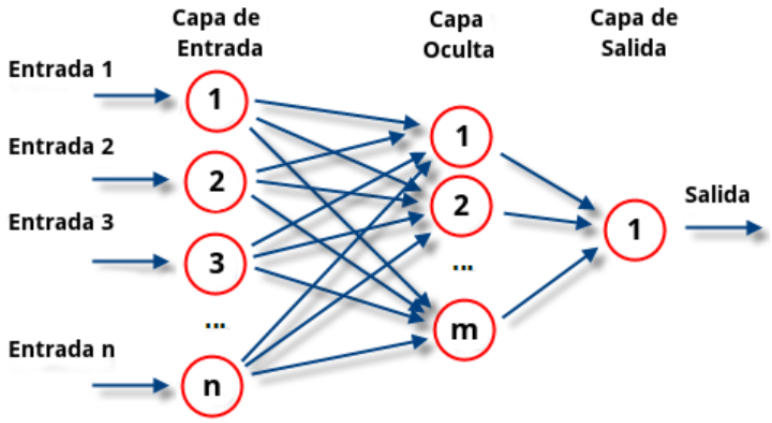


Figura 3.2: Una red neuronal y sus capas

La manera en la que las neuronas van generando sus respectivas salidas es a través de una función de activación, es la manera de transmitir la información por las conexiones de salida. La función de activación devuelve una salida que será generada por la neurona dada una entrada o conjunto de entradas. Cada una de las neuronas que conforman la red neuronal tienen una función de activación que permitirá reconstruir o predecir, es decir, obtener un resultado. La figura 3.1 muestra como la función de activación es la que produce la salida de la red.

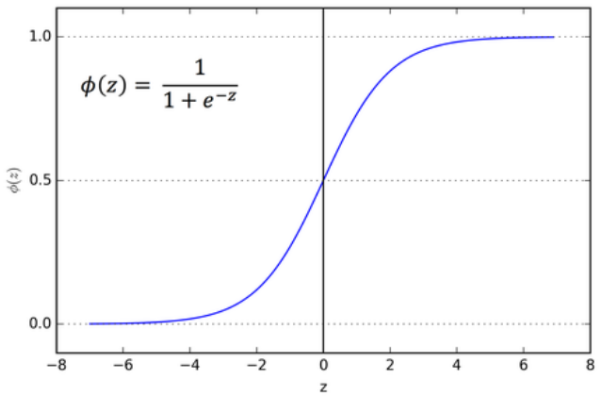
Las funciones de activación se dividen en dos tipos: lineal y no lineal. En las funciones lineales, por la misma naturaleza de dichas funciones, los resultados no estarán dados en ningún rango delimitado. Debido a esto, es recomendable para este trabajo utilizar alguna funcion de activación no lineal. Una función no lineal le facilitará al modelo generalizar o adaptarse a una mayor variedad de datos y a diferenciar las salidas.

Existen varios tipos de funciones no lineales:

1. función escalón.
2. función sigmoidal.
3. función rectificadora (ReLU).
4. función tangente hiperbólica.
5. funciones de base radial (gausianas, multicuadráticas, multicuadráticas inversas...)

Para fines de clasificación binomial las funciones de activación más apropiadas son la función sigmoidal y la función ReLu.

La función sigmoidal también conocida como función logística, está en un rango de valores de salida entre cero y uno, por lo que la salida es interpretada como una probabilidad. Si se evalúa la función con valores de entrada muy negativos, es decir x«0 la función será igual a cero, si se evalúa en cero la función dará 0.5 y en valores altos, x»0, su valor es aproximadamente a 1. Por lo que esta función se usa comunmente en problemas de clasificación binomial y al clasificar datos en dos categorías.[5] En la figura 3.3 se observa la forma de una función sigmoidal.



[5] Estefania Freie. Redes Neuronales. https://bootcampai.medium.com/redes-neuronales-13349dd1a5bb, November 2019

Figura 3.3: Función Sigmoidal y su respectiva ecuación

La función Rectified Lineal Unit, ReLu, por sus siglas en ingles, permite el aprendizaje muy rápido en las redes neuronales. Una neurona con una función de activación de ReLU toma cualquier valor

real como entrada, pero solo se activa cuando estas entradas son mayores que cero.[6] Podemos ver gráficamente una función ReLu en la figura 3.4

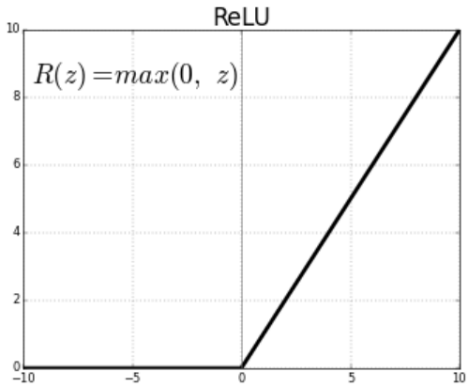


Figura 3.4: Función ReLu

Ahora se habla de otra de las funciones que se requieren en el modelo de redes neuronales, la función de pérdida. Una función de pérdida, o Loss function, es una función que evalúa la desviación entre las predicciones realizadas por la red neuronal y los valores reales de las observaciones utilizadas durante el aprendizaje. Cuanto menor es el resultado de esta función, más eficiente es la red neuronal. Su minimización, es decir, reducir al mínimo la desviación entre el valor predicho y el valor real para una observación dada, se hace ajustando los distintos pesos de la red neuronal. A continuación vemos la fórmula del error donde "real" se refiere a la predicción real y "realizada" a la predicción realizada.

$$error = |real - realizada| \tag{3.1}$$

Este error se puede considerar como error local porque se centra en una observación dada comparando el valor real y el valor predicho. [7]

Adicional a la función de pérdida y la función de activación, las redes neuronales cuentan con una función de optimización. El objetivo de ésta es mejorar la velocidad de entrenamiento y el rendimiento del modelo. En la sección de entrenamiento, se realiza el ajuste de los parametros (Hyperparameter tuning) y los pesos, para minimizar la perdida y lograr una mayor precisión de las predicciónes. El optimizador entra para modificar estos parametros, es decir que el optimizador ajusta el modelo para dejarlo de la manera mas optima al modificar los pesos del modelo, siendo la función de pérdida la que le indica al optimizador si se está moviendo hacia la dirección correcta. [8]

[6] ichi.pro. Funciones de activación: ReLU y Softmax . https://ichi.pro/es/funciones-de-activacion-relu-y-softmax-148521511786

[7] Editions ENI. Funciones de pérdida (Loss function). https://www.ediciones-eni.com/open/mediabook.aspx?idR=8dd2ca32769cb24b49648b15ef8e777e

[8] Mohit Maithani. Guide To Tensorflow Keras Optimizers . https://analyticsindiamag.com/guide-to-tensorflow-keras-optimizers/ 6, January 2021

3.3 *Árboles de Decisión*

Un árbol de clasificación o regresión es una visualización jerárquica de una serie de preguntas sobre cada unidad de la muestra. Estas preguntas se relacionan con los valores de los datos taxonómicos en cada unidad. Cuando estas preguntas son contestadas, sabremos la clase a la que pertenece cada unidad. La visualización habitual de esta información se denomina árbol debido a que es lógico representar las preguntas como un arbol al revés, con las raíces en la parte superior, una serie de ramas conectando con los nodos y hojas al final. En cada nodo se plantea una pregunta sobre una de las variables taxonómicas y la rama que se toma desde el nodo depende de la respuesta. Determinar el orden en el que se realizan estas preguntas es importante ya que determinará la estructura del arbol. Por lo general, se debe realizar aquella pregunta que maximice la ganancia en la pureza del nodo durante cada oportunidad de segmentación y en donde la pureza del nodo se mejora al minimizar la variabilidad en los datos de respuesta en el nodo. Un nodo que contenga una sola clase o categoria de la respuesta sería completamente puro [9].

[9] G. Geoffrey Vining Douglas C. Montgomery, Elizabeth A. Peck. *Introduction to Linear Regression Analysis*. John Wiley Sons Inc., sixth edition, 2021. ISBN 978-1-119-57872-7

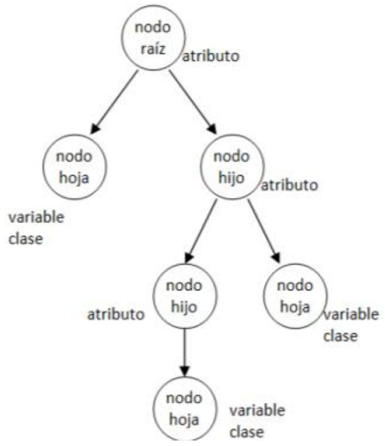


Figura 3.5: Descripción gráfica de un árbol de decisión

Dicho de otra manera, los modelos basados en árboles de decisión para clasificación se basan en estratificar o segmentar el espacio predictor en varias regiones. Estos métodos comunmente utilizan la media o la moda como valor respuesta para las observaciones de entrenamiento y asi segmentar a las distintas regiones [10].

Los árboles de decisión para clasificación se utilizan para predecir una respuesta cualitativa, a diferencia de los árboles de regresión que se utilizan para predecir respuestas cuantitavias.

[10] Trevor Hastie Rob Tishirani Gareth James, Daniela Witten. `An Introduction to Statistical Learning with Applications in R`. https://hastie.su.domains/ISLR2/ISLRv2_website.pdf, 2021

En árboles para clasificación se predice que cada observación pertence a la clase que ocurre con mayor frecuencia en la región a la que pertenece. Al interpretar los resultados de un árbol de clasificación, a menudo nos interesa no sólo la predicción de clase correspondiente a una región de nodo terminal en particular, sino también la proporción de la clase entre las observaciones de entrenamiento que caen en esa región.

Para crecer un arbol de clasificación se utilizan métodos de segmentación binarios recursivos (los segmentadores binarios son los que tienen solo dos categorías), usando una tasa de error de clasificación como criterio para hacer las divisiones. Esta tasa de error de clasificación es la fracción de observaciones de entrenamiento que no pertenecen a la clase más común y se define como:

$$E = 1 - \max_{k}(\hat{P}_{mk}) \tag{3.2}$$

en donde $\hat{P}_{mk}$ representa la proporción de observaciones de entrenamiento en la m-ésima región que forman parte la k-ésima clase.

Cuando el error de clasificación no es suficientemente sensitivo para crecer el árbol, se recomienda utilizar el ínidice de Gini o el índice de entropía. Estos dos índices se utilizan para evaluar la calidad de una partición en particular ya que ambos enfoques son más sensitivos a la pureza del nodo que el error de clasificación.

El índice de Gini mide la varianza total a través de todas las clases k, representa una media de que tan puro es el nodo, un valor pequeño indica que el nodo contiene predominantemente observaciones de una misma clase. El índice de Gini se define de la siguiente manera:

$$G = \sum_{k=1}^{K} \hat{P}_{mk}(1 - \hat{P}_{mk}) \tag{3.3}$$

Similar al índice de Gini, el índice de entropía tomará un valor cercano a cero cuando el nodo es puro. Este índice mostrará un valor cercano a cero sí los Pmk's se encuentran todos cercanos a cero o todos cercanos a uno y es dado de la siguiente manera:

$$D = -\sum_{k=1}^{K} \hat{P}_{mk} log \hat{P}_{mk} \tag{3.4}$$

Para reducir la complejidad del clasificador final y por lo tanto también mejorar la precisión del predictor al reducir el sobreajuste, se recomienda un proceso de poda. La poda del arbol de decisión, es una técnica de compresión de datos en el que algoritmos de aprendizaje autónomo reducen el tamaño del árbol al remover secciones que no son críticas o que resultan redundantes al momento de realizar las clasificaciones.

Una buena estrategia es crecer un árbol de decisión grande y despues realizar la poda y así obtener un sub-árbol. El objetivo es seleccionar un sub-árbol que conduzca a la tasa de error de prueba mas baja. Por lo que para realizar la poda se utilizan los tres métodos anteriores (tasa de error de clasificación, índice Gini y entropía), sin embargo, si el objetivo final del árbol es precisión en la predicción, utilizar la tasa de error de clasificación es preferible en el proceso de poda. [11]

[11] Trevor Hastie Rob Tishirani Gareth James, Daniela Witten. An Introduction to Statistical Learning with Applications in R. https://hastie.su.domains/ISLR2/ISLRv2_website.pdf, 2021

3·4 *Regresión Logística*

La regresión logística se utiliza en situaciónes donde la variable respuesta tiene sólo dos posibles resultados llamados éxito o fracaso y generalmente denominados por 0 y 1, siendo así esencialmente qualitativa la variable respuesta.

Generalmente, cuando la variable respuesta es binaria, la forma de la función respuesta deberia ser no lineal, creciendo o decreciendo en una función en forma de S [12]. A esta función se le llama Función Logística.

[12] G. Geoffrey Vining Douglas C. Montgomery, Elizabeth A. Peck. *Introduction to Linear Regression Analysis*. John Wiley Sons Inc., sixth edition, 2021. ISBN 978-1-119-57872-7

Suponga que el modelo tiene la forma:

$$y_i = x_i\beta + \epsilon_i \tag{3.5}$$

en donde

$$x_i = [1, x_{i1}, x_{i2}, ..., x_{ik}], \tag{3.6}$$

$$\beta = [\beta_1, \beta_2, ..., \beta_k], \tag{3.7}$$

mientras que la variable respuesta toma los valores 0 o 1. Con lo anterior podemos asumir que la variable respuesta y es una variable Bernoulli con la siguiente distribución de probabilidad:

yi	probabilidad
1	$P(yi = 1) = \pi_i$
0	$P(yi = 0) = 1 - \pi_i$

Generalmente, cuando la variable respuesta es binaria, la forma de la función respuesta debería ser no-lineal, una funcion que crece o decrece de manera monotona en forma de S. A esta función se le llama funcion logística y se muestra en la figura 3.6 y en la ecuación 3.8

$$E(y) = 1/(1 + exp(-x'\beta)) \tag{3.8}$$

Se puede observar en la Figura 3.6 que la función logística siempre va a producir una curva en forma de S cuyos valores siempre se mantendrán en un rango entre 0 y 1.

Los parámetros de un modelo de regresión logística se pueden estimar mediante el marco probabilístico denominado estimación de

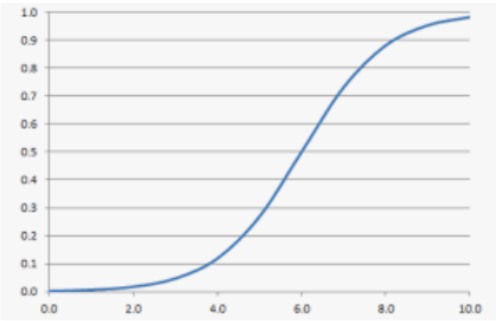


Figura 3.6: función logística

máxima verosimilitud. Bajo este marco, se debe asumir una distribución de probabilidad para la variable objetivo (etiqueta de clase) y luego definir una función de probabilidad que calcule la probabilidad de observar el resultado dados los datos de entrada y el modelo. Luego, esta función se puede optimizar para encontrar el conjunto de parámetros que da como resultado la mayor probabilidad de suma sobre el conjunto de datos de entrenamiento. Para esto es conveniente trabajar con log-verosimilitud.

$$lnL(y,\beta) = \sum_{i=1}^{n} y_i X'_i\beta - \sum_{i=1}^{n} ln[1 + exp(X'_i\beta)] \tag{3.9}$$

Una prueba de razón de verosimilitud puede usarse para comparar un modelo completo, FM por sus siglas en ingles, con un modelo reducido, RM por sus siglas en ingles. Dicho procedimiento compara dos veces el valor del logaritmo de la función de verosimilitud del modelo completo (FM) con dos veces el logaritmo del valor de la función de verosimilitud del modelo reducido (RM):

$$LR = 2[lnL(FM) - lnL(RM)] \tag{3.10}$$

La calidad del ajuste del modelo de regresión logística se puede evaluar usando una prueba de la razón de verosimilitud. Esta prueba compara el modelo actual con un modelo saturado, donde cada observación (o grupo de observaciones cuando n_i >1) se le permite tener su propio parámetro (es decir, probabilidad de éxito). Dichos parámetros o probabilidades de éxito son y_i/n_i, donde y_i es el número de éxitos y n_i el número de observaciones. La desviación es definida en log-verosimilitud como el doble de la diferencia entre el modelo saturado y el modelo original y el cual ha sido ajustado a los datos con un estimado de probabilidad de éxito π_i

$$\pi_i = \frac{exp(x_i\beta)}{1 + exp(x_i\beta)} \tag{3.11}$$

Definiendo la desviación como

$$D = 2ln\frac{L(ModeloSaturado)}{L(FM)} = 2\sum_{i=1}^{n}[y_i lm(\frac{y_i}{n_i\pi_i} + (n_i - y_i)ln(\frac{(n_i - y_i)}{n_i(1-\pi_i)})] \tag{3.12}$$

Cuando el modelo de regresión logística tiene un ajuste adecuado a los datos y el tamño de la muestra es grande, la desviación tendrá una distribución chi-cuadrada con n - p grados de libertad, donde p es el número de parámetros en el modelo. Valores pequeños de la desviación (un valor p muy alto) sugieren que el modelo provee un ajuste satisfactorio a los datos, mientras que valores altos de desviación sugieren que el modelo actual no es adecuado.

4 Desarrollo metodológico y resultados

Contents

4.1 Desarrollo metodológico con la primera base de datos y resultados

Para realizar la clasificación de las compañias entre las que incrementan su valor de las acciones en esta primer base de datos, se emplea un modelo de redes neuronales en el cual se utilizan Keras y TensorFlow. Keras, interfaz de programación para Python y una biblioteca de redes neuronales, aporta una sintaxis homogenea y una interfaz modular y expandible mientras que TensorFlow es una libreria de computación matemática open source desarrollada por Google Brain y la cual ejecuta gráficos de flujo. Para cumplir con los objetivos del presente trabajo, con la primer base de datos se implementó un modelo de redes neuronales secuencial (pila de capas secuenciales) con una capa de entrada, una capa oculta y una capa de salida. Se optó por una función de activación sigmoidal. Esta función de activación toma cualquier rango de valores a la entrada y los mapea al rango de 0 a 1 a la salida para de esta forma adecuarlos a los valores de clasificación de la variable "Class".

Para la función de pérdida, se utiliza Binary Crossentropy. Recordando que la función de pérdida tiene como objetivo medir que tan bueno es el modelo al momento de hacer predicciones. Binary crossentropy compara cada una de las probabilidades predichas con la salida real de la clase, que puede ser 0 ó 1. Luego, calcúla la puntuación que penaliza las probabilidades basándose en la distancia desde el valor esperado, eso significa qué tan cerca o lejos del valor real.

En la figura 4.1 se observa la composición de la red nueronal utilizada, vemos los 212 variables de entrada en la primer capa, una capa oculta con 8 neuronas y la capa de salida.

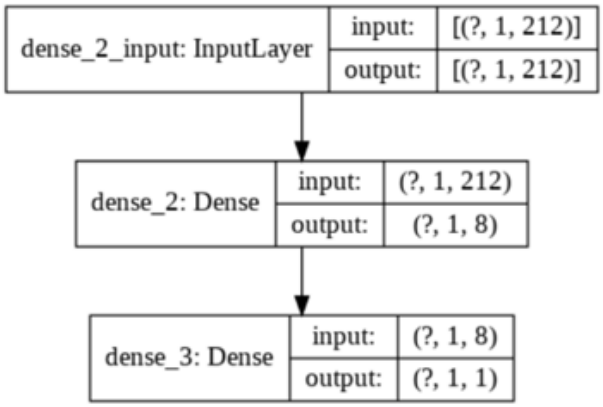


Figura 4.1: Composición de la red neuronal y sus capas

Partiendo la base de datos en 70 por ciento para el entrenamiento y 30 por ciento para probar el modelo, obtenemos los siguientes resultados del modelo implementado en la imagen 4.2. Donde accuracy mide que porcentaje de los resultados se categorizaron correctamente, mientras que la precisión muestra el porcentaje de prediciones positivas correctas y la recuperación el porcentaje de casos positivos que fueron capturados.

	Red Neuronal (dbA)		
	Accuracy	Precision	Recall
Train	0.761	0.761	0.999
Test	0.778	0.778	1.000

Figura 4.2: Resultados del modelo para dbA

4.2 *Desarrollo metodologico con la segunda base de datos y resultados*

Para esta segunda base de datos de FactSet se utilizaron tres modelos distintos de clasificación para posteriormente comparar resultados e identificar el mejor modelo para este problema: modelo de redes neuronales, modelo de arbol de decisión y un modelo de regresión logística

4.2.1 *Modelo de redes neuronales*

Este modelo, el cual tambien se construyó utlizando Keras y Tensorflow, se implementó con una capa de entrada, dos capas oculatas y una de salida. A diferencia del modelo ulitizado para la primer base de datos, se optó por la funcion de activación lineal rectificada (ReLu), como funcion de activación de las capas ocultas, esto debido a que no se recomienda utilizar las funciones de activación de la tangente sigmoidal e hiperbólica en redes con varias capas. La funcion de perdida continúa siendo Binary crossentropy.

Para el modelo aplicado a la segunda base de datos (dbB), se probaron varios ajustes y diferentes hiperparámetros con el objetivo de mejorar los resultados. Inicialmente se probó con una estructura similar a la aplicada en dbA, obteniendo resultados de exactitud y precisión menores a 50 por ciento, por lo que el primer paso fue incrementar el tamaño de la base de datos, de tener información de 50 compañias se pasó a 200 compañias. Posteriormente, se probó cambiando la función de activación de la capa oculta, identificando que la función ReLu mejoraba estos parámetros en varios puntos porcentuales. Se probó tambien añadiendo una capa oculta adicional con distintas cantidades de neuronas.

Partiendo la base de datos en 70 por ciento para el entrenamiento y 30 por ciento para probar el modelo, obtenemos los siguientes resultados del modelo implementado:

	Red Neuronal (dbB)		
	Accuracy	Precision	Recall
Train	0.908	0.895	0.958
Test	0.662	0.674	0.721

Figura 4.3: Resultados del modelo para dbB

4.2.2 *Modelo de árbol de decisión*

El segundo modelo aplicado a la base de datos fue un modelo de árbol de decisión para clasificación. Se buscaron los mejores parámetros utilizando la herramienta GridSearchCV. Dicha herramienta encuentra y ajusta automáticamente los parámetros a través de un método de busqueda de cuadrícula y validación cruzada, combinando todos los parámetros especificados para encontrar la que de como resultado el mejor modelo. GridSearchCV arrojó como resultado la siguiente combinación:

1. Medición de pureza (Función para medir la pureza o calidad de la división, GridSearchCV puede tomar Gini o Entropy): Entropy.
2. Divisor (Método utilizado para la división en cada nodo, pudiendo adoptar "Best"(Mejores) para elegir la mejor división o Random"(aleatorio) para elegir la mejor división aleatoria): Best.
3. Estado aleatorio (Controla la aleatoriedad del estimador): 8.
4. Máxima profundidad (La profundidad máxima del árbol): 5.
5. Máxima característica (La cantidad de características a considerar al buscar la mejor división): Auto.

Este modelo de árbol obtuvo los siguintes resultados de entrenamiento y prueba:

	Arbol de decision		
	Accuracy	Precision	Recall
Train	0.775	0.828	0.857
Test	0.975	1.000	0.953

Figura 4.4: Resultados modelo arbol de decisión

A continuación, se analiza la matriz de confusión mostrada en la figura 4.5, mostrando que el modelo clasificó incorrectamente como falsos positivos al 12.5 porciento de los datos, mientras que existieron 10 por ciento clasificados como falsos negativos.

Finalmente, se realizó un análisis de selección de características. La selección de características es el proceso de identifiar las variables más relevantes para así reducir el número de variables de entrada al desarrollar un modelo predictivo.

Es deseable reducir el número de variables de entrada para reducir el costo computacional del modelado y, en algunos casos, para mejorar el rendimiento del modelo.

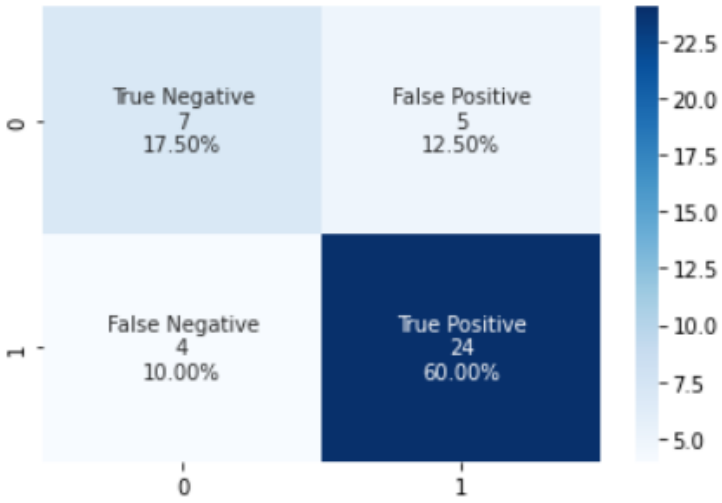


Figura 4.5: Matriz de confusión: Arbol de decisión

Los métodos de selección de características basados en estadísticas implican evaluar la relación entre cada variable de entrada y la variable de destino utilizando estadísticas y seleccionando aquellas variables de entrada que tienen la relación más fuerte con la variable de destino.[1]

Al realizar la selección de caraterísticas obtenemos los siguientes resultados mostrados en la figura 4.6. En los cuales observamos que las variable más significativa es SP 500 seguida por Industry, Div Yld y Asset Turn(turover).

Tomando en cuenta este resultado, se ejecutó de nuevo el arbol de decisión con solo dichas variables representativas (SP 500, Industry, Div Yld Asset Turn). Con esto, se obtuvieron los resultados que se muestran en la figura 4.7, y en la cual podemos observar un overfitting en el train, lo que sugiere que el modelo no funcionaría solo con estas variables representativas.

[1] Jason Brownlee. How to Choose a Feature Selection Method For Machine Learning. https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/, 2019

4.2.3 *Modelo de regresión logística*

El tercer y último modelo aplicado a la base de datos fue un modelo de regresión logística. Para este modelo se utilizó el módulo de LogisticRegression de SkLearn, partiendo los datos en 80 por ciento para entrenamiento y 20 por ciento para prueba, arrojando los siguientes resultados (Figura 4.8) y matriz de confusión (Figura 4.9):

4.3 *Discusión*

Para evaluar los resultados del modelo de clasificación, se utilizan las metricas de exactitud (accuracy), precisión y exhaustividad o

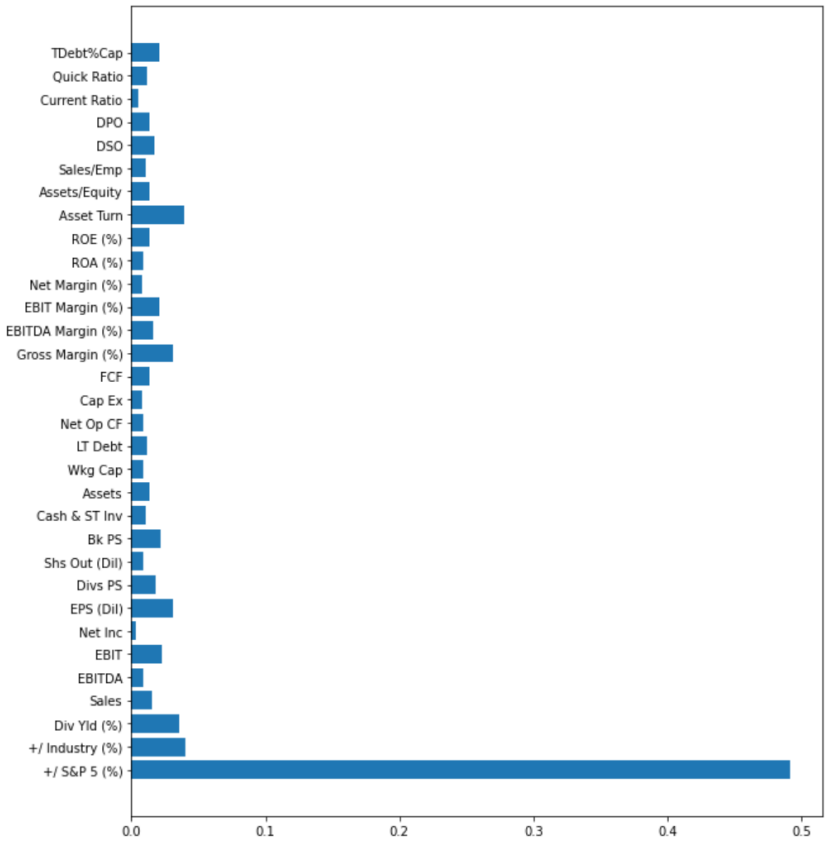


Figura 4.6: Importancia de las variables

Árbol de decisión (4 Variables)			
	Accuracy	Precision	Recall
Train	1	1	1
Test	0.675	0.810	0.654

Figura 4.7: Resultado Arbol de Decisión - 4 Variables

Regresión Logística			
	Accuracy	Precision	Recall
Train	0.669	0.750	0.833
Test	0.650	0.700	0.808

Figura 4.8: Resultados modelo regresión logística

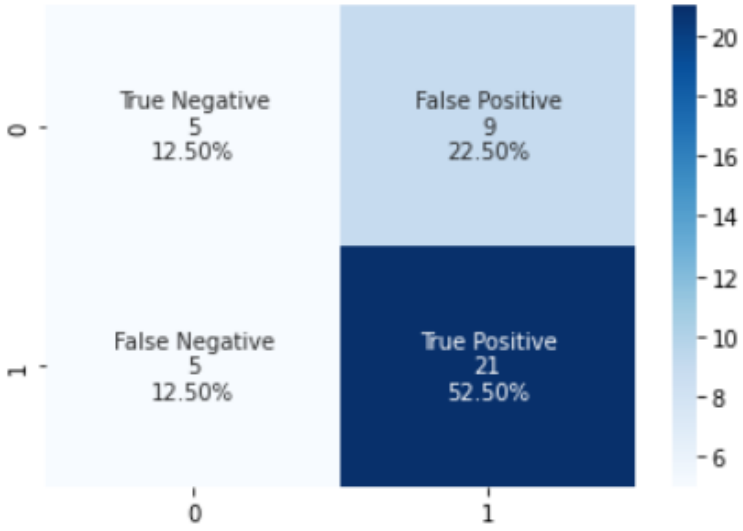


Figura 4.9: Matris de confusión: Regresión logística

recuperación (recall).

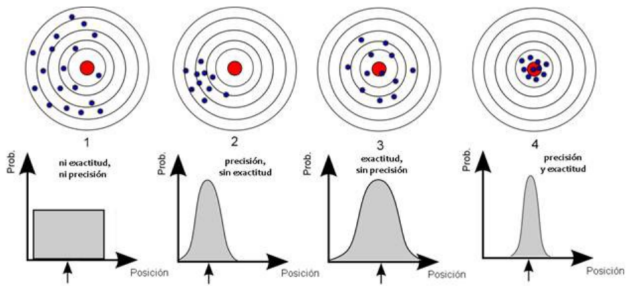


Figura 4.10: Graficas de exactitud y precisión

Para poder obtener estas métricas, el modelo debe realizar una matriz de confusión en donde se plasme lo siguiente:

1. True Negative (TN): Caso cuando es negativo y es predecido negativo.
2. True Positive (TP): Caso cuando es positivo y es predecido positivo.
3. False Negative (FN): Caso cuando es positivo, pero es predecido negativo.
4. False Positive (FP) : Caso cuando es negativo, pero es predecido positivo.

Posteriormente se aplican las siguientes formulas para obtener los resultados de precisión y recuperación:

precisión = True Positive / (True Positive + False Positive)

recuperación = True Positive / (True Positive + False Negative)

4.3.1 *Red neuronal: Comparación de resultados entre ambas bases de datos*

Los resultados del modelo aplicado a cada base de datos reflejan algunas diferencias en exactitud y precisión. La primera muestra que es posible clasificar las acciones con un 76.1 por ciento de accuracy y un 79.4 por ciento de precisión, mientras que la segunda reduce de manera importante el resultado de accuracy y precision a un 66.2 y un 67.4 por ciento respectivamente. Importante mencionar el nivel de recuperación de la primer base de datos es del 100 por ciento.

Se observa entonces que el primer modelo arroja niveles de mayor precisión y accuracy lo cual se puede deber a varios factores entre los que se encuentran:

1. Mayor cantidad de datos en la base de datos que arrroja resultados satisfactorios. La primer base de datos cuenta con infomración de 1502 compañias (despues de la limpieza de la base de datos) y 216 variables, mientras que la segunda base de datos cuenta solo con 200 compañias y 34 variables. Esto pudiera ser el principal factor determinante debido a la diferencia significativa de datos entre una y otra base de datos.

2. Valores faltantes. Mientras que la primer base de datos contenía una cantidad mayor de valores faltantes, existe la posibilidad que el metodo empleado de promedios por variable y sector influyera o sesgara el modelo dando resultados satisfactorios falsos.

4.3.2 *Resultados de modelos aplicados a la segunda base de datos*

Como se mencionó anteriormente, se aplicaron tres distintos modelos de clasificación a la base de datos dbB con el fin de poder comparar y analizar los distintos resultados arrojados por cada modelo.

Se observan los mejores resultados en los dos modelos mas simples, arbol de decisión y regresión logística, mientras que los resultados de la red neuronal son deficientes, arrojando por debajo del 0.7 tanto accuracy como precisión. Esto nos indica que sería poco conventiente utilizar este modelo para clasificar nuestra variable de salida.

El modelo de regresión logística refleja resultados, pricipalmente en Precisión y Recall, sin embargo, con un Accuracy y precisión menores a 0.8 vemos que tampoco sería correcto utilizar este modelo para nuestra base de datos, ya que podría arrojar resultados incorrectos.

Finalmente, el arbol de decisión muestra una mejora significativa en los resultados de prueba comparando con los otros dos modelos. Vemos los tres parámetros por arriba del 0.95. Al revisar los resultados de la matriz de confusión vemos una clasificacion correcta, tanto en true positive como en true negative del 78 por ciento de los datos. Esto indica que el modelo más adecuado de los tres implementados, es el de

árbol de decisión. Regresando a la matriz de confusión de este modelo, el falso positivo arroja un 12.5 por ciento, el cual es el porcentaje mas pequeño comparando contra los otros modelos, si bien es importante obtener altos porcentajes en los tres parametros, este trabajo pone especial atención en los Falsos positivos, ya que una clasificación de una compañia u acción como falso positivo podria resultar en invertir en una compañia que no estaría creciendo su valor año con año.

5 Conclusiones y trabajo futuro

Contents

A continuacion se abordarán las conclusiones con base a los resultados obtenidos y el trabajo futuro de la presente tesis.

5.1 Conclusiones

En base a los resultados obtenidos de aplicar el modelo de redes neuronales para clasificación a la primer base de datos, se observa un buen desempeño al analizar las metricas de exactitud, precisión y recuperación. El modelo para la primer base de datos refleja una posibilidad de acercarse a la clasificación correcta mediante solo los resultados financieros de las compañias.

Al momento de usar este modelo para realizar pruebas futuras será importante mantener en consideración el método de llenado de valores faltantes, el cual pudiera estar sesgando los resultados.

Los resultados del modelo de red renuronal aplicado a la segunda base de datos son moderados, esto reflejado tambien por las metricas de exactitud, precisión y recuperación con resultados dentro del 60 al 70 por ciento.

Sabemos que las complejidades que mueven los precios de las acciones en el mercado de valores son amplias, no son solo los resultados financieros de las compañias los que determinan el precio, esto es solo uno de los factores importantes. Otros factores ya mencionados son: especulación, variables economicas externas, cambios organizacionales, etc. Estos factores pudieran ser tambien influencia importante para que el modelo aqui presentado resulte con metricas moderadas de precisión.

Como se menccionó en la sección de discusión, el modelo de arbol de decisión es el que obtiene las mejores metricas de accuracy, precisión y recall tanto en el entrenamiento como en el test.

5.2 *Trabajo futuro*

El trabajo futuro de este proyecto se centrará en buscar incrementar la base de datos dbB para que así los modelos de red neuronal, árbol de decisión y regresión logística tengan más información para trabajar al expandir el conjunto de entrenamiento. A la vez que se buscarán distintos métodos de llenado de valores faltantes.

Adicional al punto anterior, se revisarán modelos de ensamble aplicados a estas bases de datos, buscando asi mejorar el rendimiento al mejorar la precesión. En los modelos de ensamble los datos se suministran a un conjunto de modelos, y luego se combinan las predicciones de los modelos.

Bibliografía

U.S. Securities and Exchange Commission. https://www.sec.gov/edgar.shtml, January 2017.

PLLC . Anthony L.G. Public Company SEC Reporting Requirements . http://www.legalandcompliance.com/securities-resources/sec-requirements-for-public-companies/.

Jaime Ariel-Toral Barrera. Redes Neuronales. http://www.cucei.udg.mx/sites/default/files/pdf/toral_barrera_jamie_areli.pdf.

Yang Bai. Machine Learning Classification Methods and Portfolio Allocation: An Examination of Market Efficiency. https://herbert.miami.edu/_assets/pdfs/faculty-research/business-conferences/machine-learning/yang-bai.pdf, January 2021.

Jason Brownlee. How to Choose a Feature Selection Method For Machine Learning. https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/, 2019.

IBM Corporation. El modelo de redes neuronales. https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=networks-neural-model.

G. Geoffrey Vining Douglas C. Montgomery, Elizabeth A. Peck. *Introduction to Linear Regression Analysis*. John Wiley Sons Inc., sixth edition, 2021. ISBN 978-1-119-57872-7.

Editions ENI. Funciones de pérdida (Loss function). https://www.ediciones-eni.com/open/mediabook.aspx?idR=8dd2ca32769cb24b49648b15ef8e777e.

Nicolás Muñoz y Edwin García Fernando Villada. Aplicación de las Redes Neuronales al Pronóstico de Precios en el Mercado de Valores. https://scielo.conicyt.cl/scielo.php?pid=S0718-07642012000400003&script=sci_arttext&tlng=en, November 2012.

Estefania Freie. Redes Neuronales. https://bootcampai.medium.com/redes-neuronales-13349dd1a5bb, November 2019.

Trevor Hastie Rob Tishirani Gareth James, Daniela Witten. An Introduction to Statistical Learning with Applications in R. https://hastie.su.domains/ISLR2/ISLRv2_website.pdf, 2021.

Abhishek Gupta. A Survey on Stock Market Prediction Using Various Algorithms . https://www.academia.edu/43646075/A_Survey_on_Stock_Market_Prediction_Using_Various_Algorithms, March 2014.

ichi.pro. Funciones de activación: ReLU y Softmax . https://ichi.pro/es/funciones-de-activacion-relu-y-softmax-148521511785096.

Kaggle. 200+ Financial Indicators of US stocks (2014-2018)). https://www.kaggle.com/cnic92/200-financial-indicators-of-us-stocks-20142018.

Yun Li. 80 percent of the stock market is now on autopilot . https://www.cnbc.com/2019/06/28/80percent-of-the-stock-market-is-now-on-autopilot.html, June 2019.

AMG Funds LLC. Fundamental vs. Technical Analysis. https://tinyurl.com/7bmb8zk.

Mohit Maithani. Guide To Tensorflow Keras Optimizers . https://analyticsindiamag.com/guide-to-tensorflow-keras-optimizers/6, January 2021.

Financial Modelling Prep. Financial Modelling Prep. https://site.financialmodelingprep.com/developer/docs.

Runebook.dev. sklearn.preprocessing.MinMaxScaler. https://runebook.dev/es/docs/scikit_learn/modules/generated/sklearn.preprocessing.minmaxscaler.

sec.gov. La SEC: Lo que Somos, y lo que Hacemos. https://www.sec.gov/investor/espanol/quehacemos.htm, September 2001.

Simplilearn. Everything You Need to Know About Classification in Machine Learning. https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning, November 2021.

Shivam Sinha. Making Machine Learning Work For Financial Market Prediction. https://www.

forbes.com/sites/forbesfinancecouncil/2021/10/18/making-machine-learning-work-for-financial-market-prediction/?sh=1c5e5a8e2cce, October 2021.

Jin Liu Sohrab Mokhtari, Kang K. Yen. Effectiveness of Artificial Intelligence in Stock Market Prediction based on Machine Learning. https://arxiv.org/abs/2107.01031, June 2021.

Alvira Swalin. How to Handle Missing Data. https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4, January 2018.

Índice alfabético