

Universidad Autónoma de Querétaro

Facultad de Ingeniería

Título:

Optimización de Portafolios de Inversión en la Bolsa Mexicana de Valores Utilizando
Redes Neuronales Artificiales.

Tesis que como parte de los requisitos para obtener el grado de:

Maestro en Ciencias con línea terminal en Ingeniería Matemática

Presenta:

Guillermo Carlos Guerrero Camargo

Dirigido por:

Dr. Víctor Antonio Aguilar Arteaga

Centro Universitario, Querétaro, Qro.

Septiembre 2022.

México



Dirección General de Bibliotecas y Servicios Digitales
de Información



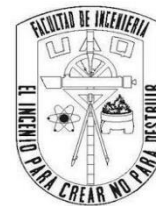
Optimización de Portafolios de Inversión en la Bolsa
Mexicana de Valores Utilizando Redes Neuronales
Artificiales

por

Guillermo Carlos Guerrero Camargo

se distribuye bajo una [Licencia Creative Commons
Atribución-NoComercial-SinDerivadas 4.0
Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Clave RI: IGMAC-214370



Universidad Autónoma de Querétaro
Facultad de Ingeniería

Título:

Optimización de Portafolios de Inversión en la Bolsa Mexicana de Valores Utilizando
Redes Neuronales Artificiales.

Tesis que como parte de los requisitos para obtener el grado de:

Maestro en Ciencias con línea terminal en Ingeniería Matemática

Presenta:

Guillermo Carlos Guerrero Camargo

Dirigido por:

Dr. Víctor Antonio Aguilar Arteaga

Dr. Víctor Antonio Aguilar Arteaga (Presidente) _____

M en C. Iván González García (Secretario) _____

Dr. Samuel Estala Arias (Vocal) _____

Dr. Francisco Gerardo Jiménez López (Suplente) _____

M en C. Samuel Joseph Lizarazu Cerón
(Suplente) _____

Centro Universitario, Querétaro, Qro.
Septiembre 2022.
México

Dedicatorias

Dedico el proyecto a Dios por darme la oportunidad de concluir una etapa más, a mi mamá y a mi papá que han sido y son un soporte incondicional en el día a día; también a la Universidad Autónoma de Querétaro que me prestó las instalaciones para desarrollarme como profesionista.

Agradecimientos

Agradezco al CONACYT por el apoyo para estudiar, así como a la UAQ por todos los recursos prestados, al profe Víctor; que me asesoró en esta tesis y durante la maestría, por todas las pláticas de economía y de matemáticas las cuales me ayudaron a mejorar mi conocimiento profesional. A mis profesores que aportaron algo constructivo en mi profesión.

Contenido

Resumen.....	1
Abstract.....	2
Introducción.....	3
Antecedentes.....	4
Hipótesis.....	10
Objetivos.....	10
Objetivo General.....	10
Objetivos particulares:.....	10
Fundamentación teórica.....	11
Selección de portafolios de inversión.....	11
Incertidumbre económica.....	11
Aprendizaje supervisado (SL).....	11
Aprendizaje no supervisado (UL).....	11
Aprendizaje por refuerzo (RL).....	12
Aprendizaje automático (ML).....	12
Estimación.....	12
Clasificación.....	12
Aprendizaje profundo (DL).....	12
Aprendizaje incremental.....	12
Aproximación universal.....	13
Redes neuronales.....	13
Volumen y Variedad.....	13
Precios de arbitraje.....	13
Teoría de la utilidad esperada.....	14
Teoría de la cartera de la varianza media.....	14
Relación de Sharpe.....	14
El modelo de valoración de activos de capital.....	14
La teoría de precios de arbitraje.....	14
Tarea de aprendizaje.....	14
Teoría de un portafolio.....	14
Eficiencia del mercado al final del día.....	15
Evitar el sobreajuste.....	15
Predicción del mercado (portafolio) usando redes neuronales artificiales.....	15

Mercados Eficientes.....	16
Manera débil de la HME.....	16
Forma semi fuerte de la HME.	16
Forma fuerte de la HME.	16
Predicción de mercado basada en datos de rentabilidad.	18
Las ineficiencias estadísticas	18
Predicción de mercado intradía.....	18
Utilidad de los negocios intradiarios.	18
Riesgo y rendimiento.	19
Conjunto factible y conjunto eficiente.	19
El portafolio de mínimo riesgo.....	20
Con n activos financieros.....	21
Riesgo y rendimiento.	21
Metodología.....	25
Optimización del portafolio de inversión de H. Markowitz. (Teoría de portafolios)	26
Usar RNA para analizar un portafolio de inversión.....	27
Analizar de manera individual usando RN, con los siguientes puntos.....	27
Resultados y discusión.....	37
Teoría del portafolio óptimo	37
Análisis individual (por cada empresa) usando redes neuronales artificiales.....	43
Datos.....	43
Éxito	48
Capacidad	56
Evaluación.	62
Sesgo y varianza	70
Validación cruzada	73
Mercados eficientes	74
Predicción de mercado basada en datos de devoluciones.....	75
Predicción de mercado con más características.	77
Predicción de mercado intradía.....	79
Conclusiones.....	84
Bibliografía	85

Resumen

Hoy en día la mercadotecnia ha fomentado el interés en las inversiones, porque desde un teléfono inteligente es fácil usar aplicaciones que permiten hacer inversiones en tiempo real. Dentro de la estrategia de inversión se encuentra un proceso para invertir dinero, buscando siempre obtener el mayor rendimiento al menor riesgo posible; de esta manera se resuelve el problema de como seleccionar portafolios de inversión. La primera persona que planteó el problema de la selección de portafolios de inversión como un problema matemático fue Harry Markowitz; quien, en la década de 1950; publicó un modelo que permite optimizar los portafolios de inversión. Realizar investigación pública en finanzas, especialmente en la selección de portafolios de inversión; contribuye al aprendizaje de los interesados para fortalecer el conocimiento del comportamiento de los negocios y con esto, tomar decisiones solidas en el armando de carteras financieras. Teniendo como referencia el modelo del economista H. Markowitz, junto con el uso de redes neuronales artificiales para desarrollar una estructura que ayude a ampliar, estudiar y analizar un portafolio de inversión que cotiza en la Bolsa Mexicana de Valores en los años 2016 al 2022; a pesar del uso actual de redes neuronales artificiales en los negocios, continuar con el estudio de estas, sirve para encontrar formas nuevas de cómo usarlas en el análisis del mercado. El comportamiento de la economía es complejo, existen severas variables que son parte de los cambios constantes de precios que no se pueden controlar, la intención de indagar en esta área es para inferir el comportamiento de la bolsa; no para controlarla, sino para determinar el riesgo que implica invertir, haciendo uso de técnicas matemáticas; al final; el inversionista tiene la última palabra, derivado del análisis de carteras. También mencionar; las redes neuronales son útiles, debido a la capacidad de aprendizaje que tienen; con esto, ayudaran a mejorar la selección de portafolios, para tener el mejor rendimiento con el mínimo riesgo con ayuda del programa Python 3.6.

Palabras clave: Optimización, Cartera, Inversión, Rendimiento.

Abstract

Today, marketing has encouraged interest in investments, because from a smartphone it is easy to use applications that allow investments in real time. Within the investment strategy there is a process to invest money, always seeking to obtain the highest return at the lowest possible risk; in this way the problem of how to select investment portfolios is solved. The first person to pose the problem of investment portfolio selection as a mathematical problem was Harry Markowitz; who, in the 1950s; published a model that allows optimization of investment portfolios. Carry out public research in finance, especially in the selection of investment portfolios; contribute to the learning of the interested parties to strengthen the knowledge of the behavior of the businesses and with this, make solid decisions in the construction of financial portfolios. Taking as a reference the model of the economist H. Markowitz, together with the use of artificial neural networks to develop a structure that helps to expand, study, and analyze an investment portfolio that is listed on the Mexican Stock Exchange in the years 2016 to 2022; Despite the current use of artificial neural networks in finance, continuing to study them serves to find new ways to use them in market analysis. The behavior of the economy is complex, there are several variables that are part of the constant changes in prices that cannot be controlled, the intention of investigating in this area is to lower the behavior of the stock market; not to control it, but to determine the risk involved in investing, using mathematical techniques; at the end; the investor has the last word, derived from portfolio analysis. I also mention; neural networks are a useful tool, due to their learning capacity; with this, they will help to improve the selection of portfolios, to have the best performance with the minimum risk with Python 3.6 software.

Keywords: Optimization, Portfolio, Investment, Performance.

Introducción

Los métodos de optimización pueden ser mejorados y no existe un modelo ideal para seleccionar un portafolio de inversión, que maximice el rendimiento con un mínimo de riesgo, así que, aunque existan modelos de optimización de carteras de inversión para la Bolsa Mexicana, estos modelos deben estudiarse constantemente, pues las condiciones políticas, económicas y sociales son dinámicas en el tiempo, además día con día se tiene mejor infraestructura para calcular grandes cantidades de información. Estudiar el modelo de optimización dicho por Harry Markowitz, es un buen ejercicio de introducción al tema; para poder hacer mejoras en vías de obtener mejores resultados, con base a las necesidades del inversor.

Además, se busca analizar el portafolio de inversión, utilizando redes neuronales artificiales, estas muestran un excelente desempeño en otras áreas de la ciencia. Este trabajo de tesis pretende contribuir de manera positiva, a la persona que desee iniciarse en la optimización de portafolios de inversión para la Bolsa Mexicana de Valores, utilizando herramientas matemáticas, pues además de presentar, formular y resolver el modelo Harry Markowitz, se muestra cómo se puede hacer una adecuación del modelo agregando herramientas de Inteligencia Artificial, con esto se calcula todos los datos históricos de valores de las acciones de forma rápida y, con un nivel de precisión. Estudiar un portafolio de inversión implica la posibilidad de mayores ganancias económicas o evitar una mala inversión.

Antecedentes.

En 1938, John Burt Williams publicó el libro *“The Theory of Investment Value”*, en este libro plantea que, para calcular el precio de una acción, se empieza por anticipar cuántas divisiones se reportarán, después; acomodar la predicción a la inflación, intereses y otras variables que provocan incertidumbre. En 1950, Harry Markowitz; alumno de PhD en Economía, en una universidad de Chicago, se encontraba leyendo el libro de John Burt Williams mientras buscaba tema de tesis, y no considero que fuera correcta la estrategia descrita anteriormente. Esto es que, Markowitz pensó que los inversionistas no consideraban sólo su beneficio potencial. Si así lo hicieran, la mayoría compraría acciones de la compañía que mejores ganancias reporta. En vez de eso Markowitz observó que la gente tiende a diversificar sus inversiones para mitigar el riesgo. Es así como Markowitz planteó su problema de tesis doctoral: establecer un modelo matemático para optimizar un portafolio de inversión siguiendo un antiguo dicho “no introduces todos los huevos en la misma canastilla”. Así Markowitz fue la primera persona en plantear y resolver, el problema de optimizar un portafolio de inversión utilizando herramientas propias de las matemáticas [1] [2].

En el artículo de R. Lawrence, titulado: *“Using Neural Networks to Forecast Stock Market Prices”* se examinó el uso de las redes en los sistemas financieros. Demostrando la eficiencia de las redes que se han utilizado para probar la hipótesis del mercado eficiente, cómo superan las estadísticas y técnicas de regresión para pronosticar los precios de las acciones. Menciona que, aunque las redes neuronales no son perfectas en su predicción, superan a todos los demás métodos y brindan la esperanza de que algún día se pueda comprender mejor los sistemas dinámicos y caóticos como el mercado de valores [3].

En la revista de J. Del Carpio, *“Las Redes Neuronales Artificiales en las Finanzas”*, menciona que la RNA es una ecuación matemática, elaborada de una cantidad grande de números de conjuntos administrados en capas. La RNA, es vista como un sistema de procedimiento computacional, hecho por una cantidad de factores unidos quedan seguimiento a los datos, cambiando los resultados dinámicos, dando lugar a factores externos.

La aplicación de las RNA en la economía es diferente. Se encuentran en la gestión de carteras, evaluación de bonos y acciones, atajos de arbitraje, estudio de la predicción, correlación, varianza de los costos, tasa de interés y cambio. La información debe estar sujeta a principios sólidos: Primero los datos deben conseguirse de los mercados. Se debe preparar la información de salida y entrada. El seguimiento de identificación de la información necesitada está en función del asunto que se esté estudiando; de esa manera se dice que la RNA que pueda adivinar algún tipo serie de tiempo de factores económicos, puede construir usando diferentes tipos de información. Aun así, de lo difícil del aprendizaje de las Redes Neuronales, el análisis de la literatura permite mostrar su amplia ejecución en el tema económico, en particular, en la predicción de variables financieras. Esto tiene que ver con el pronóstico de crisis, rendimientos, que mejores la toma de decisiones. Con esto, las Redes Neuronales pueden usarse como utensilios complementarios en las maneras comunes de análisis multivariante [4].

Hablando de la tesis de A. Álvarez, A. Trillos, *“Apoyo a la Investigación: Estructuración de Portafolios de Inversión en Acciones Usando Redes Neuronales; Uso de los Métodos Tradicionales”*; se mencionó que al desarrollar un conjunto de redes neuronales auto organizadas por Kohonen con el programa de Matlab, no fue favorable el resultado, porque los datos no tienen muchas características que los diferencie y por tal motivo al momento de agrupar la red, este no los ordena lógicamente y todos los datos los agrupa en un mismo apartado; a raíz de estos resultados optaron por otra red neuronal artificial llamada perceptrón multicapa. La red entrenada (la multicapa) les arrojó resultados favorables. Llenó sus expectativas, pues con la teoría de portafolios de Markowitz, les dio resultados parecidos. Al realizar el portafolio con las acciones que les arrojó la red, vieron que era más favorable porque dio como resultado un grupo de acciones más rentables y por esta razón es recomendable. Mirando los demás modelos para la estructuración de portafolio, encontraron que el modelo APT no es usado en un sistema económico colombiano porque su mercado de capital es muy pequeño, solo tiene una bolsa de mercado accionario [5].

En la revista de O. García, A. Morales “*Las Redes Neuronales Artificiales como una herramienta de Análisis en la Determinación de las Empresas que Permanecen Listadas o Des listadas dentro de la Bolsa Mexicana de Valores*”, menciona que, es un análisis que, mediante la RNA, decide qué negocios de la economía mexicana continuarán cotizando o, no estarán participando. Esto con el interés de dar herramientas a expertos para la elaboración con carteras que estudian el riesgo y ganancias, dando por entendido que, más adelante una empresa no continúe en la lista, provocaría movimientos a estos portafolios, teniendo consecuencias diferentes a las metas organizadas al inicio. Lo previo, en este encuentro, se determinó que, las causas financieras y señales bursátiles son importantes para las acciones en la BMV para que se mantengan listadas o no. La revisión checó un conjunto de 21 empresas dentro de las listas y 11 no, analizando cada una, 21 argumentos financieros y dos señalamientos para el intervalo del año 2005 al 2010. El resumen fue que esta investigación del desarrollo neuronal que se consiguió fue estructurada a 23:4:2, precisando el 97.3 % para seleccionar a los negocios en las listas. Los resultados obtenidos con estos métodos de RNA enseñaron la técnica con una capacidad alta de selección y predicción para la posible permanencia de las empresas en la BMV. Sobre la búsqueda se halló que uno de tantos sistemas neuronales es la *MLP 23-4-2*, con una recepción derivada con cinco motivos (primera: motivo de movimiento, segunda: que tan rentable, otra de generar valores y un indicador bursátil), que genera que la RNA seleccione que empresas seguirán en el mercado. Agregando, el sistema sale de un nivel escondido con 4 uniones o nodos ocultos, teniendo actividades de colocación estratégicas y Softmax, y una retirada con dos uniones o nodos que, seleccionan dentro o fuera del listado, calibrando la estancia en la BMV. La destreza de la RNA tiene una capacidad para seleccionar cerca del 97.3 %, tomada en cuenta alta. Concluyendo, se dice sobre que, el arte de las redes ayuda a economistas, en la calibración de la estancia de las emisoras que se venden en la Bolsa, un utensilio diferente al método común que brinde alta certidumbre para la elección de selección involucradas con la Bolsa [6].

Hablando de la revista de E. Carmona, *“Ajuste a la Calificación del Riesgo de Mercado de las Acciones más Volátiles que Conforman el Índice de Precios y Cotizaciones de la Bolsa Mexicana de Valores, con la Implementación de una Red Neuronal Artificial Clasificadora”*, menciona que, el análisis está relacionado con que, no existen actividades anteriores acerca de búsquedas en la comparación de varianza o riesgo del mercado mexicano, haciéndose una especie de RNA Clasificador con el fin de motivar el encuentro de nuevos conjuntos en la selección; diversos análisis se concentran a la comparación de rentabilidad para las acciones. Comenta que, se inicia un nuevo camino para promover la búsqueda en el ramo económico, dando una estrategia para acomodar la comparación de lo no deseado o sea el riesgo, economía mexicana, por lo tanto, al usar RNA como utensilios para selección de riesgo de la economía o del mercado, los involucrados tendrán una opción que los guíe en la búsqueda de sus negocios para elevar sus ganancias, mejorando las decisiones. Concluyendo, con la estructura, se enseña el interés de las RNA como una máquina para calibrar y seleccionar la varianza de la cartera mexicana, aplicándolo a otras áreas económicas. La información de inicio que se introdujeron a la RNA eran elegidos del sistema de VAC (Valuación de Activos de Capital) y la información final bien fundamentada por las teorías clásicas, para comparar el riesgo del mercado. Viendo anteriormente, el procedimiento de la RNA elaborado con los intereses de la búsqueda, se solidifica al mezclar un sistema económico que es lógico y de IA, tal como lo son las RNA; para esto el ejemplo utilizado, está hecho por información de crisis económicas como la que ocurrió en el 2004, que golpeó de manera significativa en los índices mundiales de las finanzas, pero crisis del año 2008 dañó a países de primer mundo[7].

En este artículo de J. Frausto, E. Álvarez, J. Sánchez, P. Limonchi y J. Lebrun, “*Análisis de Portafolios de Inversión para la Bolsa Mexicana de Valores*”, dice que, la responsabilidad de un analista financiero consiste en estudiar alternativas para la toma de decisiones. Para evaluar el riesgo se requiere de una intuición desarrollada con el apoyo de modelos derivados de la estadística, optimización y métodos heurísticos, que consideran el sentido común. El riesgo no sólo involucra incertidumbre, sino también oportunidades de inversión. La incertidumbre es información que no puede ser determinada por completo y es difícil de cuantificar, mientras que el riesgo es medible mediante las probabilidades de ocurrencia o estimadores como lo es la varianza y covarianza. El hecho de que no sea deseado el riesgo no hace que sea fácil medir.

Mencionando también qué, la Retro propagación, conocido también como *Back propagation* (por su designación en inglés) es el método de redes neuronales (ANN, de sus siglas en inglés Artificial Neural Networks) más utilizado por la mayoría de los investigadores de ANN y en particular para aplicaciones de series financieras. Sin embargo, esta técnica es en realidad un método de entrenamiento de Redes Neuronales que se emplea reservando un parte del conjunto de datos que conforman la serie de tiempo financiera y que se conoce como conjunto de entrenamiento. Aunque, esta técnica produjo al principio resultados prometedores, se sabe que no funciona del todo bien. Las ANN tienen un gran defecto que la comunidad científica señala; tienden a quedar atrapado en óptimos locales. Las ANN, a partir del conjunto de entrenamiento, determinan con frecuencia de manera muy rápida el modelo financiero, pero no siempre de forma correcta. Como resultado, si tal método es aplicado para anticipar la variación es probable que esta no sea ni siquiera de forma aproximada con el número posterior real, ya que el asunto de mejora locales puede aparecer. Se han usado diferentes enfoques para lidiar de resolver ese problema; sin éxito hasta ahora. Este artículo de “Análisis de Portafolios de Inversión para la Bolsa Mexicana de Valores” se muestra un chequeo de las técnicas de estudio de volatilidad aplicado al armado de carteras óptimas de inversión para la Bolsa Mexicana de Valores. Tales carteras se rigen por medio de reducir el riesgo sistemático de las herramientas de búsqueda, se busca un análisis de las diferentes variables de mejora enseñando que estrategias heurísticas que manejen aleatoriedad son una opción correcta para dicho problema. Se estudian las técnicas

no iguales de análisis de riesgos, afrontando lo histórico de tales métodos. Se señala que las estrategias de percepción de información son los más indicados para la anticipación del cambio, los nombrados Máquinas de Soporte Vectorial (son series para procedimientos de aprendizaje supervisado). Esta actividad muestra una visión resumida del alto objetivo que representa el resultado óptimo de las carteras de inversión [8].

En el libro de N. Mancilla, M. Muñoz, M. Sánchez, *“Estructuración de portafolios mediante el uso de redes neuronales: un comparativo de la rentabilidad frente a los métodos tradicionales”*; hace mención de la elaboración de carteras, por medio de las redes neuronales; esto mostró ser una técnica útil, mostrando; tanto a desarrollar la ganancia producida con otras estructuras aplicadas, así que; tuvo mejor variación, ayudado por las acciones que se incluyeron en la cartera y disminución en el nivel de duda a través del uso de RNA para adelantarse a cambios futuros. Siendo lo mismo, el interés de indicar con la aplicación de redes, para la predicción con líneas de intervalo de tiempo financieras necesita un alto nivel de empeño; aun así, de que en el análisis se hallaron diversos comentarios para el desarrollo de la red, tuvieron que corroborar todas las mezclas disponibles por medio de MATLAB, hasta hallar lo mejor con cada línea de tiempo de información conseguida. De igual manera, encontraron esta estructura metodológica que, no sirve con predicciones ni con carteras a mediano y largo plazo, ya que cada intervalo de tiempo predicho estará sujeto a la anterior lógica, por lo cual el error incrementa con relación al tiempo. Por este punto comentaron qué, se sugiere acomodar los intervalos de información de acuerdo con la predicción que se necesite hacer. De la misma manera, utilizar esta estrategia en mercados con alta diferencia de acciones; se dice que, se sugiere calibrar esta estrategia propuesta con 40 y otros de estructuración de carteras, con el fin aplicar más panorama a lo conseguido, conseguidos en la actual investigación [9].

En resumen, el presente trabajo reúne las características anteriormente mencionadas por distintos autores, mencionando las RNA en las finanzas, aportando nuevo conocimiento y dando a conocer, cómo se usan para realizar actividades económicas.

Hipótesis.

Las Redes Neuronales Artificiales tendrán una precisión del 80% de predicción en el estudio de un portafolio de inversión en la Bolsa Mexicana de Valores.

Objetivos.

Objetivo General

Demostrar la precisión de predicción usando Redes Neuronales Artificiales en el análisis un portafolio de inversión derivado de la Bolsa mexicana de Valores del año 2016 hasta el 2022.

Objetivos particulares:

1. Optimizar un portafolio de inversión usando las herramientas de Harry Markowitz usando Python.
2. Utilizar redes neuronales artificiales para predecir el comportamiento de un portafolio de inversión derivado de la Bolsa Mexicana de Valores.

Fundamentación teórica.

Selección de portafolios de inversión.

Se habla de "selección de cartera" en lugar de "selección de valores". Un buen portafolio no solo es una lista de excelentes acciones. Es un equilibrio, que aporta al economista opciones contra diferentes mercados. El inversionista debe hacer un portafolio que se ajuste a sus objetivos y el interés del estudio es buscar los portafolios que se ajusten a los intereses del inversionista. Lo dicho por Markowitz ayuda a demostrar lo que se menciona como el límite eficiente o frontera, lo cual se construye como el grupo de carteras unidas por varias mezclas de rendimiento y riesgo que se pueden tener entre muchas acciones y que brindan una ganancia alta [1].

Incertidumbre económica.

La duda es una condición sobresaliente de un negocio; dado que los factores económicos no se comprenden un tanto bien, para que la predicción este más allá del error. A pesar, si los resultados de las exigencias financieras se entendieran perfectamente, las influencias no financieras tienen opción de modificar el trayecto del bienestar total, la capa económica o el éxito de un valor puntual [1].

Aprendizaje supervisado (SL).

Estos son algoritmos que aprenden de un conjunto de datos de muestra dada, de valores de características (entrada) y etiquetas (salida); ejemplos de dichos algoritmos: la regresión de mínimos cuadrados ordinarios (OLS) y las redes neuronales. El propósito del aprendizaje supervisado es captar la unión entre los datos de salida y entrada. En finanzas, dichos algoritmos pueden entrenarse para predecir si un deudor potencial es solvente o no, estos son los tipos de algoritmos más importantes [10].

Aprendizaje no supervisado (UL).

Estos son algoritmos que aprenden de un conjunto de datos de muestra dada, solo de valores de características (entrada), a menudo con el objetivo de encontrar una estructura en los datos. Se supone que deben aprender sobre el conjunto de datos de entrada; por ejemplo, algunos parámetros de guía. Los algoritmos de agrupamiento entran en esa categoría. En un contexto financiero, tales algoritmos podrían agrupar acciones en ciertos grupos [10].

Aprendizaje por refuerzo (RL).

Estos son algoritmos que aprenden de prueba y error, al recibir una recompensa por realizar una acción. Actualizan una política de actuación óptima según las recompensas y castigos que reciben. Dichos algoritmos se utilizan, por ejemplo, para entornos en los que es necesario realizar acciones continuamente y las recompensas se reciben de inmediato, como en un juego de computadora [10].

Aprendizaje automático (ML).

ML es la disciplina de relaciones de aprendizaje e información sobre conjuntos de datos dados basados en un algoritmo y una medida de éxito; una medida del éxito podría ser; por ejemplo, el error cuadrático medio (MSE), dados los valores de las etiquetas y los valores de salida que se estimarán y los valores predichos del algoritmo. ML es un subconjunto de la AI [10].

Estimación.

La estimación (o aproximación, regresión) se refiere a los casos en los que los datos de las etiquetas tienen un valor real (continuo); es decir, técnicamente se representa como números de punto flotante [10].

Clasificación.

La clasificación se refiere a los casos en los que los datos de las etiquetas consisten en un número finito de clases o categorías, que normalmente se representan mediante valores discretos (números naturales), que a su vez se representan técnicamente como números enteros [10].

Aprendizaje profundo (DL).

DL engloba todos los algoritmos basados en redes neuronales. El término profundo generalmente solo se usa cuando la RNA tiene diversas capas ocultas. DL es un subconjunto de aprendizaje automático y, por lo tanto, también es un subconjunto de AI [10].

Aprendizaje incremental.

Los pesos óptimos dentro de una red neuronal, dada una medida objetivo de éxito, se aprenden de forma incremental en función de una inicialización aleatoria y mejoras incrementales. Estas mejoras incrementales se logran al considerar la resta con los números antes dichos, los de etiqueta de muestra y las actualizaciones de ponderaciones retro propagadas a través de la red neuronal [10].

Aproximación universal.

Existen teoremas matemáticos que muestran que las RNA (incluso una capa) pueden aproximarse a casi cualquier función [10].

Redes neuronales.

Las redes neuronales son buenas para aprender la interconexión entre la salida y entrada. Se pueden aplicar a varios tipos de problemas, como la estimación en presencia de relaciones complejas o la clasificación, para los que los métodos estadísticos tradicionales no suelen ser los adecuados [10].

Volumen y Variedad.

En el contexto de las redes neuronales que realizan tareas de predicción, el volumen y la variedad de la información a disposición que se toman para adiestrar la red neuronal, son determinantes para su desempeño predictivo. Los ejemplos hipotéticos numéricos muestran que la misma red neuronal entrenada en un conjunto de datos relativamente pequeño y no tan variado, tiene un rendimiento inferior a su contraparte entrenada en un conjunto de datos relativamente grande y variado en más de 10 puntos porcentuales. Esta diferencia puede considerarse enorme dado que los profesionales de la IA y las empresas a menudo luchan por mejoras tan pequeñas como una décima de punto porcentual [10].

Precios de arbitraje.

La hipótesis de colocación de valores o precios de arbitraje puede considerarse una de las teorías financieras más sólidas con algunos de los resultados matemáticos más sólidos, como el teorema fundamental de fijación de precios de activos (FTAP). Entre otras razones, esto se debe al hecho de que los precios de las opciones, por ejemplo, pueden derivarse de otros parámetros de mercado observables, como, el precio de las acciones sobre las que se escribe la opción. La fijación de precios de arbitraje en ese sentido no se ocupa de cómo llegar a un precio de acción justo en primer lugar, sino que simplemente lo toma como una entrada. Por lo tanto, la fijación de precios de arbitraje ya funciona con pocas y leves suposiciones, como la ausencia de arbitraje, lo que no se puede decir de muchas otras teorías financieras. Tenga en cuenta que ni siquiera se utiliza la medida de probabilidad para derivar el precio de arbitraje [10].

Teoría de la utilidad esperada.

Esta hipótesis (EUT) asume que los economistas consiguen una función similar de ganancia o utilidad sin importar el estado del mundo y que maximizan la utilidad esperada en condiciones de incertidumbre [10].

Teoría de la cartera de la varianza media.

Esta hipótesis (MVP) describe cómo los inversionistas deben invertir en condiciones de incertidumbre, suponiendo que solo cuentan el rendimiento y la volatilidad esperados de una cartera durante un período [10].

Relación de Sharpe.

Sharpe introduce una medida para juzgar la ganancia modificando la varianza no deseada de los montos mutuos y otras carteras, o incluso activos de riesgo individuales. En su forma más simple, relaciona el rendimiento (esperado, realizado) de una cartera con su volatilidad (esperada, realizada) [10].

El modelo de valoración de activos de capital.

El modelo de fijación de precios de activos de capital (CAPM), asume que solo el riesgo de mercado no diversificable explica el rendimiento y la volatilidad esperados de una acción durante un período [10].

La teoría de precios de arbitraje.

APT asume que una serie de factores de riesgo identificables, explican el rendimiento y la volatilidad esperados de una acción a lo largo del tiempo; es cierto que, en comparación con las otras teorías, la formulación de APT es bastante amplia y permite interpretaciones de gran alcance [10].

Tarea de aprendizaje

En la definición de aprendizaje a través de un algoritmo, es importante señalar la tarea de aprendizaje. Aprender significa entender cómo (mejorar) ejecutar una determinada tarea, como la estimación o la clasificación. [10]

Teoría de un portafolio

Empezando por saber la teoría de un portafolio para estar entendiendo lo que se pretende mostrar al público, conociendo las preferencias del inversionista al explicar que considera al rendimiento esperado como algo deseable y a la varianza del rendimiento como algo indeseable. El modelo propuesto por Harry Markowitz una de las teorías más conocidas sobre

la selección de portafolios, con base a la frecuencia de la ganancia media y con la varianza del rendimiento, entendiendo esta última como una representación del riesgo, aunque en términos de desviación estándar. Una cartera o un portafolio de acciones financieras es una forma de inversión en la que se distribuye la totalidad del recurso invertido en los diferentes activos que lo conforman. El portafolio eficiente resulta de un proceso de minimizar el riesgo maximizando el rendimiento; mientras que el Sharpe ratio es maximizar el rendimiento.

Los principales supuestos en los que está basado el modelo son:

1. Teoría de los mercados eficientes.
2. Los rendimientos de los activos continúan la distribución normal.
3. Se sabe el promedio, la varianza y las correlaciones de los rendimientos. [10]

Eficiencia del mercado al final del día.

La hipótesis del mercado eficiente se remonta a las décadas de 1960 y 1970, períodos durante los cuales los datos al final del día eran básicamente los únicos datos de series temporales disponibles. En aquellos días (y todavía hoy), se podía suponer que los actores del mercado prestaban especial atención a sus posiciones y operaciones a menudo de aproximaba el término de la etapa de negociación. [10]

Evitar el sobreajuste.

El sobreajuste, cuando un ejemplo económico sirve mejor con una diversidad de información de adiestramiento que en los conjuntos de valores de prueba y validación, debe evitarse en el ML y en finanzas en particular. Los procedimientos y análisis de evaluación adecuados, como la validación cruzada, ayudan a prevenir el sobreajuste y a encontrar, por ejemplo; una capacidad de modelo adecuada [10].

Predicción del mercado (portafolio) usando redes neuronales artificiales.

Esta parte combina las finanzas basadas en datos con el enfoque de aprendizaje automático del procedimiento anterior. Las redes neuronales se utilizan para descubrir ineficiencias estadísticas en los “negocios eficientes o mercados”, analizando la teoría de los negocios eficientes y usando la regresión de OLS para ilustrar con base a los datos de series temporales financieras. “Predicción del mercado basada en datos de rendimiento”, aplicando redes neuronales, junto con la regresión OLS, para adivinar la dirección anticipada del valor de una herramienta financiera (“dirección del mercado”). El análisis se basa únicamente en datos de

retorno. “Predicción de mercado con más funciones”, agregando más funciones a la combinación, como indicadores financieros típicos. En este tema, los primeros resultados indican que las ineficiencias estadísticas podrían estar presentes. Esto se confirma en “Predicción del mercado intradía (dentro del día)”, que funciona con datos intradía en comparación con los datos al final del día [10].

Mercados Eficientes.

Una de las hipótesis con mayor apoyo empírico es la hipótesis del mercado eficiente (HME), también se denomina teoría de los pasos cambiantes o aleatorios (HCA). En pocas palabras, la teoría dice que, los valores de instrumentos financieros sobre un determinado instante proyectan los datos a disposición sobre ese momento. Si la HME es cierta, una discusión sobre si el valor de un activo es mucho más bajo o alto no tendría sentido. El precio de una acción, dada la HME, se encuentra en todo momento exactamente en su nivel adecuado a la información disponible. Se ha puesto mucho esfuerzo en refinar y formalizar la idea de mercados eficientes desde la formulación y las primeras discusiones de la HME en la década de 1960 todavía se utilizan en la actualidad. Jensen define un mercado eficiente de la siguiente manera; un conjunto de negocios es certero en referencia a un cumulo de datos θ_t , si es; inviable conseguir oportunidades financieras tratando sobre el cimientto del cumulo de datos θ_t . Por beneficios económicos entendemos los rendimientos ajustados al riesgo netos de todos los costes [10].

En este contexto, Jensen distingue tres formas de eficiencia de mercado;

Manera débil de la HME.

En este caso, el conjunto de información θ_t solo abarca el historial de precios y retornos pasados del mercado [10].

Forma semi fuerte de la HME.

En este caso, se consideró que el monto de datos θ son todos los valores a disposición públicamente, que incluye no solo el historial de precios y retornos pasados, sino también informes financieros, artículos de noticias, datos meteorológicos, etc. [10].

Forma fuerte de la HME.

Este caso se da cuando el monto de datos θ son todos los valores a disposición para cualquier persona (es decir, incluso la información privada). No importa qué forma se adopte, las implicaciones de la HME son de largo alcance. Durante muchos años, los economistas,

estadísticos y profesores de finanzas se han interesado en desarrollar y probar modelos de actividad de valores de activos. Un modelo importante que ha evolucionado a partir de esta investigación es la teoría de los paseos aleatorios. Esta teoría arroja serias dudas sobre muchos otros métodos para describir y adivinar la actividad de los activos, métodos que tienen una considerable popularidad fuera del mundo académico [10].

En otras palabras, si la HME es cierta, cualquier tipo de investigación o análisis de datos siendo el fin de conseguir ganancias optimas al mercado debería ser inútil en la práctica. Por otro lado, ha evolucionado una industria de gestión de activos multimillonaria que promete rendimientos superiores al mercado debido a una investigación rigurosa y la gestión activa del capital. En particular, la industria de fondos de cobertura se basa en promesas de entregar alfa, es decir; rendimientos superiores al mercado e incluso independientes, al menos en gran medida, de los rendimientos del mercado. Los datos de un estudio reciente de Preqin muestran cuán difícil es cumplir con esa promesa. El estudio informa una caída en el índice *Preqin* Fondo de cobertura de todas las estrategias de -3,42% para el año 2018. Cerca del 40% de todos los fondos de cobertura cubiertos por el estudio, experimentaron pérdidas del 5% o más para ese año. Si el valor de un activo (o el valor de cualquier otro instrumento financiero) sigue un paseo aleatorio estándar, los rendimientos se distribuyen normalmente con media cero. El precio de las acciones sube con un 50% de probabilidad y baja con un 50% de probabilidad. En tal contexto, el mejor predictor del precio de las acciones de mañana, en un sentido de mínimos cuadrados, es el precio de las acciones de hoy. Esto se debe a la propiedad de Markov de los paseos aleatorios, es decir, que la distribución de valores futuros de los activos es independiente de la trayectoria del proceso de costos; solo depende del nivel de precios actual. Por lo tanto, en el contexto de un paseo aleatorio, el análisis de los precios (o rendimientos) históricos es inútil para predecir precios futuros. En este contexto, se puede implementar una prueba semiformal para mercados eficientes de la siguiente manera. Tome una serie de tiempo financiera, retrase los datos de precios varias veces y use los datos de precios rezagados como datos de características para una regresión OLS que usa el nivel de precios actual como los datos de las etiquetas. Esto es similar en las técnicas de gráficos que se basan en formaciones históricas de precios para predecir valores futuros. Los siguientes pasos implementa un análisis de este tipo basado en datos de precios

rezagados para una serie de instrumentos financieros, tanto negociables como no negociables [10].

Predicción de mercado basada en datos de rentabilidad.

Los algoritmos del ML y, en particular el DL, han generado avances en los últimos años en campos que han demostrado ser resistentes durante largos períodos de tiempo a los métodos estadísticos o matemáticos estándar. ¿Qué pasa con los mercados financieros? ¿Podrían los algoritmos del ML y DL ser capaces de descubrir ineficiencias donde fallan los métodos tradicionales de econometría financiera, como la regresión OLS? Por supuesto, todavía no hay respuestas simples y concisas a estas preguntas [10].

Las ineficiencias estadísticas

Se dan cuando un modelo es capaz de predecir la dirección del movimiento futuro del precio con cierta ventaja (por ejemplo, la predicción es correcta en el 55 % o el 60 % de los casos). Las ineficiencias económicas solo se darían si las ineficiencias estadísticas pueden explotarse de manera rentable a través de una estrategia comercial que tenga en cuenta, por ejemplo, los costos de transacción [10].

Predicción de mercado intradía.

Este tema no ha producido evidencia concluyente, pero los análisis implementados hasta ahora apuntan más en la dirección de que los mercados son poco eficientes al final del día. ¿Qué pasa con los mercados intradiarios? ¿Hay ineficiencias estadísticas más consistentes que detectar? Para trabajar hacia una respuesta a esta pregunta, se necesita otro monto de información. El proceso utiliza un monto de información que se compone con los mismos instrumentos que en el conjunto de datos al final del día, esto contiene precios de cierre por hora [10].

Utilidad de los negocios intradiarios.

Incluso si los negocios son poco útiles al final del día, pueden no obstante ser débilmente ineficientes en modo “intradía”. Tales ineficiencias estadísticas pueden resultar de desequilibrios temporales, presiones de compra o venta, reacciones exageradas del mercado, órdenes de compra o venta impulsadas técnicamente, etc. La pregunta central es si tales ineficiencias estadísticas, una vez descubiertas, pueden usarse de manera adecuada a través de estrategias comerciales específicas [10].

Riesgo y rendimiento.

El conjunto factible una cartera de activos como ejemplo de dos acciones se obtiene representando gráficamente las posibles combinaciones del riesgo y el rendimiento esperado. Sea P_t y P_{t-1} los precios de un activo financiero para los periodos t y $t - 1$, y sea $R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$ su correspondiente rendimiento, entonces se define el rendimiento esperado del activo para una muestra de T periodos, como Ec.1.

$$RE(R_t) = \bar{R} = \frac{1}{T} \sum_{t=1}^T R_t \quad \text{Ec (1).}$$

Mientras que el riesgo, Ec.2.

$$RI(R_t) = DE(R_t) = \sqrt{\text{Var}(R_t)} = \sqrt{\frac{1}{T-1} \sum_{t=1}^T (R_t - \bar{R})^2} \quad \text{Ec (2).}$$

Se tienen los activos financieros 1 y 2; y las respectivas proporciones invertidas en cada uno de ellos w_1 y w_2 , es posible construir un portafolio de inversión $P(w_1, w_2)$, donde $w_1 + w_2 = 1$

Por lo anterior, el rendimiento esperado de $P(w_1, w_2)$ es $RE(P) = w_1\bar{R}_1 + w_2\bar{R}_2$ y su respectivo riesgo, Ec.3.

$$RI(P) = \sqrt{w_1^2 \text{Var}R_1 + 2w_1w_2 \text{Cov}(R_1R_2) + w_2^2 \text{Var}R_2} \quad \text{Ec (3).}$$

Conjunto factible y conjunto eficiente.

Dado que la varianza del portafolio es: $\text{Var}(P) = w_1^2 \text{Var}R_1 + 2w_1w_2 \text{Cov}(R_1R_2) + w_2^2 \text{Var}R_2$ el papel de la covarianza en el riesgo del portafolio es determinante.

Valores positivos de la covarianza generan un mayor riesgo del portafolio, valores negativos generan un menor riesgo.

Por lo anterior, la curva del conjunto factible será más pronunciada mientras más negativa sea la covarianza y viceversa.

Dado que el coeficiente de correlación ρ entre los dos activos, está dado por, Ec.4.

$$\rho(R_1R_2) = \frac{\text{Cov}(R_1,R_2)}{DE(R_1)DE(R_2)}, -1 \leq r \leq 1 \quad \text{Ec (4)}$$

Mientras menor sea ρ mayor será la curva del conjunto factible y viceversa.

El portafolio de mínimo riesgo.

El portafolio de mínima varianza se llama portafolio óptimo.

El portafolio óptimo se encuentra eligiendo las proporciones invertidas en cada uno de los activos financieros, w_1 y w_2 , tal que la varianza del portafolio sea la mínima.

El problema de mejorar con el fin de encontrar la cartera con el mínimo riesgo con dos activos financieros, si se permite la venta en corto, está dado por, Ec.5.

$$\text{Min } w_1^2 \text{Var}R_1 + 2w_1w_2 \text{Cov}(R_1R_2) + w_2^2 \text{Var}R_2 \quad \text{Ec (5)}$$

$$\text{Sujeto a } w_1 + w_2 = 1$$

Sean $s_1^2 = \text{Var}(R_1)$, $s_2^2 = \text{Var}(R_2)$ y $s_{1,2} = \text{Cov}(R_1R_2)$, la función de Lagrange está dada por, Ec.6.

$$\mathcal{L} = w_1^2 s_1^2 + 2w_1w_2 s_{1,2} + w_2^2 s_2^2 + \lambda(1 - w_1 - w_2) \quad \text{Ec(6)}$$

Mientras que, las condiciones de primer orden son, Ec.7, 8 y 9.

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1 s_1^2 + 2w_2 s_{1,2} - \lambda = 0 \quad \text{Ec (7)}$$

$$\frac{\partial \mathcal{L}}{\partial w_2} = 2w_1 s_{1,2} + 2w_2 s_2^2 - \lambda = 0 \quad \text{Ec (8)}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 1 - w_1 - w_2 = 0 \quad \text{Ec (9)}$$

Al resolver las ecuaciones de la 7 a la 9 se obtiene los siguientes valores para las proporciones invertidas en cada uno de los activos financieros, Ec 10 y 11.

$$w_1 = \frac{s_2^2 - s_{1,2}}{s_1^2 + s_2^2 - 2s_{1,2}} \quad \text{Ec (10)}$$

$$w_2 = 1 - w_1 \quad \text{Ec (11)}$$

El resultado anterior se vuelve a expresar de la siguiente forma, Ec.12.

$$w_1 = \frac{\text{Var}(R_2) - \text{Cov}(R_1, R_2)}{\text{Var}(R_1) + \text{Var}(R_2) - 2\text{Cov}(R_1, R_2)} \quad \text{Ec (12)}$$

$$w_2 = 1 - w_1 \quad \text{Ec (11)}$$

Para encontrar el portafolio para dos activos financieros, se plantea el siguiente problema, Ec.13.

$$\text{min} = w_1^2 \text{Var}R_1 + 2w_1w_2 \text{Cov}(R_1R_2) + w_2^2 \text{Var}R_2 \quad \text{Ec (13)}$$

Sujeto a, Ec.14.

$$w_1 + w_2 = 1, w_1 \geq 0, w_2 \geq 0 \quad \text{Ec (14)}$$

Con n activos financieros.

Ahora como ejemplo; si se tienen los activos financieros 1, 2, 3 ..., n y las respectivas proporciones invertidas en cada uno de ellos $w_1, w_2, w_3, \dots, w_n$ es posible construir un

portafolio de inversión $P = w_1, w_2, w_3, \dots, w_n = \mathbf{P}$

Donde $w_1 + w_2 + w_3, \dots, +w_n = 1$

Riesgo y rendimiento.

Para presentar de forma sencilla los conceptos de ganancia esperada y varianza no deseada o riesgo sobre n activos se utilizan matrices.

Sea \mathbf{w} el vector (n por 1) de las n proporciones, \mathbf{w}^T el vector transpuesto (1 por n) de las n proporciones, Ω la matriz (n por n) de las varianzas y covarianzas y $\bar{\mathbf{r}}$ el vector (n por 1)

de los n rendimientos, el rendimiento esperado del portafolio \mathbf{P} está dado por $RE(\mathbf{P}) = \mathbf{w}^T \bar{\mathbf{r}}$

Y el correspondiente riesgo por, Ec.15.

$$RI(\mathbf{P}) = \sqrt{\mathbf{w}^T \Omega \mathbf{w}} \quad \text{Ec (15)}$$

Al igual que para un portafolio con dos activos, es posible crear un conjunto de portafolios para diferentes combinaciones de las proporciones invertidas en cada uno de los activos financieros. Por dicha combinación de portafolios es posible encontrar un conjunto factible y un conjunto eficiente.

En la siguiente gráfica; figura 1, se muestran 5,000 portafolios para cinco activos financieros, en donde puede apreciarse el conjunto factible, el conjunto eficiente y el portafolio con el mínimo riesgo.

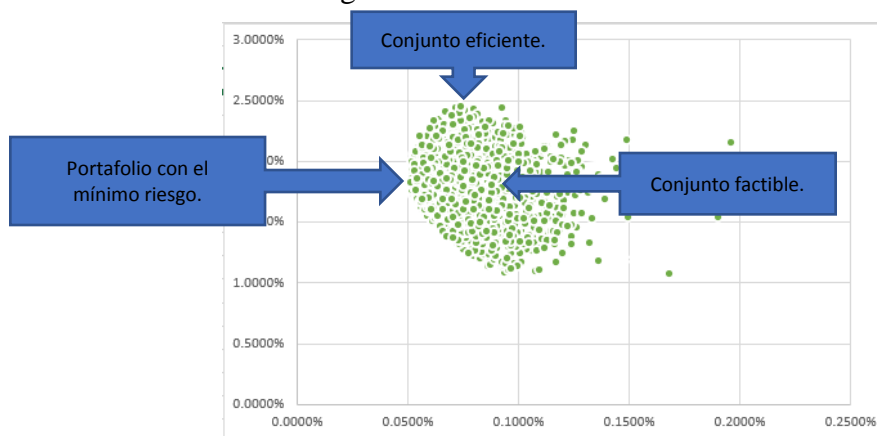


Figura 1. Ejemplo de grafica con 5000 portafolios.

Para encontrar numéricamente la frontera eficiente o algunos portafolios sobre ella, permitiendo la venta en corto, se plantea el siguiente problema

Minimizar $\mathbf{w}^T \boldsymbol{\Omega} \mathbf{w}$ y Minimizar $\mathbf{w}^T \boldsymbol{\Omega} \mathbf{w}$

Sujeto a: Ec. 16 y 17

$$\bar{r}_p = \mathbf{w}^T \bar{\mathbf{r}} \quad \text{Ec (16)}$$

$$1 = \mathbf{w}^T \mathbf{1}_n \quad \text{Ec (17)}$$

Donde $\mathbf{1}_n$ es un vector n por 1.

La solución al problema anterior está dada por: Ec.18.

$$\mathcal{L} = \mathbf{w}^T \boldsymbol{\Omega} \mathbf{w} + \lambda_1 (\mathbf{w}^T \bar{\mathbf{r}} - \bar{r}_p) + \lambda_2 (\mathbf{w}^T \mathbf{1}_n - 1) \quad \text{Ec (18)}$$

Con las condiciones de primer orden: Ec. 19, 20 y 21

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 2\boldsymbol{\Omega} \mathbf{w} + \lambda_1 \bar{\mathbf{r}} + \lambda_2 = \mathbf{0}_n \quad \text{Ec (19)}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_1} = \mathbf{w}^T \bar{\mathbf{r}} - \bar{r}_p = 0 \quad \text{Ec (20)}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_2} = \mathbf{w}^T \mathbf{1}_n - 1 = 0 \quad \text{Ec (21)}$$

Multiplicando las dos caras de la igualdad (19) por $\frac{1}{2}$ y despejar los términos independientes de las ecuaciones (20) y (21), el sistema puede expresarse matricialmente de la forma, Ec.22.

$$\begin{bmatrix} \boldsymbol{\Omega} & \bar{\mathbf{r}} & \mathbf{1}_n \\ \bar{\mathbf{r}}^T & 0 & 0 \\ \mathbf{1}_n^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \lambda_1/2 \\ \lambda_2/2 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_n \\ \bar{r}_p \\ 1 \end{bmatrix} \quad \text{Ec (22)}$$

O, despejando matricialmente, Ec.23.

$$\begin{bmatrix} \mathbf{w} \\ \lambda_1/2 \\ \lambda_2/2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Omega} & \bar{\mathbf{r}} & \mathbf{1}_n \\ \bar{\mathbf{r}}^T & 0 & 0 \\ \mathbf{1}_n^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0}_n \\ \bar{r}_p \\ 1 \end{bmatrix} \quad \text{Ec (23)}$$

De la expresión anterior, la variable exógena es \bar{r}_p , por lo que, para encontrar los diferentes portafolios que conforman la frontera eficiente basta con asignarle los valores que se desean alcanzar y encontrar el conjunto de ecuaciones.

Para encontrar numéricamente la frontera eficiente o algunos portafolios sobre ella, permitiendo la venta en corto, se plantea el siguiente problema

Minimizar $\mathbf{w}^T \boldsymbol{\Omega} \mathbf{w}$

Sujeto a; Ec.24, 25 y 26.

$$\bar{r}_p = \mathbf{w}^T \bar{\mathbf{r}} \quad \text{Ec (24)}$$

$$1 = \mathbf{w}^T \mathbf{1}_n \quad \text{Ec (25)}$$

$$\mathbf{w} \geq \mathbf{0}_n \quad \text{Ec (26)}$$

Para encontrar la solución a este problema es recomendable utilizar algún algoritmo que lo resuelva de forma iterativa. Los resultados serán similares a los encontrados cuando se permite la venta en corto, solo que ahora todos los pesos son positivos.

Para encontrar el portafolio óptimo de Markowitz con n activos financieros, permitiendo la venta en corto, se plantea el siguiente problema

Minimizar $\mathbf{w}^T \Omega \mathbf{w}$

Sujeto a; Ec.27.

$$1 = \mathbf{w}^T \mathbf{1}_n \quad \text{Ec (27)}$$

La solución estaría dada por; Ec.28.

$$\mathcal{L} = \mathbf{w}^T \Omega \mathbf{w} + \lambda (\mathbf{w}^T \mathbf{1}_n - 1) \quad \text{Ec (28)}$$

Con las condiciones de primer orden; Ec. 29 y 30

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 2\Omega \mathbf{w}_0 + \lambda = \mathbf{0} \quad \text{Ec (29)}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \mathbf{w}^T \mathbf{1}_n - 1 = 0 \quad \text{Ec (30)}$$

Donde \mathbf{w}_0 representa el portafolio de mínimo riesgo.

Al multiplicar la ecuación (4) por $\frac{1}{2}$ y despejar el termino independiente de (5) se tiene; Ec.31.

$$\begin{bmatrix} \Omega & \mathbf{1}_n \\ \mathbf{1}_n^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_0 \\ \lambda/2 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_n \\ 1 \end{bmatrix} \quad \text{Ec (31)}$$

La expresión anterior se despeja para \mathbf{w}_0 y se obtiene; Ec.32.

$$\begin{bmatrix} \mathbf{w}_0 \\ \lambda/2 \end{bmatrix} = \begin{bmatrix} \Omega & \mathbf{1}_n \\ \mathbf{1}_n^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0}_n \\ 1 \end{bmatrix} \quad \text{Ec (32)}$$

Por lo anterior, la ganancia esperada y la varianza no deseada para la cartera \mathbf{P} conformado por las proporciones \mathbf{w}_0 están dados por; Ec.32 y 33.

$$RE (P) = \mathbf{w}_0^T \bar{\mathbf{r}} \quad \text{Ec (32)}$$

$$RI (P) = \mathbf{w}_0^T \Omega \mathbf{w}_0 \quad \text{Ec (33)}$$

Del resultado anterior puede observarse que, para construir el conjunto eficiente, los

valores que deben tomar \bar{r}_p deben estar por encima del rendimiento esperado del portafolio de mínimo riesgo.

Si se desea encontrar el portafolio óptimo para n activos, sin permitir la venta en corto, se plantea el siguiente problema

Minimizar $\mathbf{w}^T \Omega \mathbf{w}$

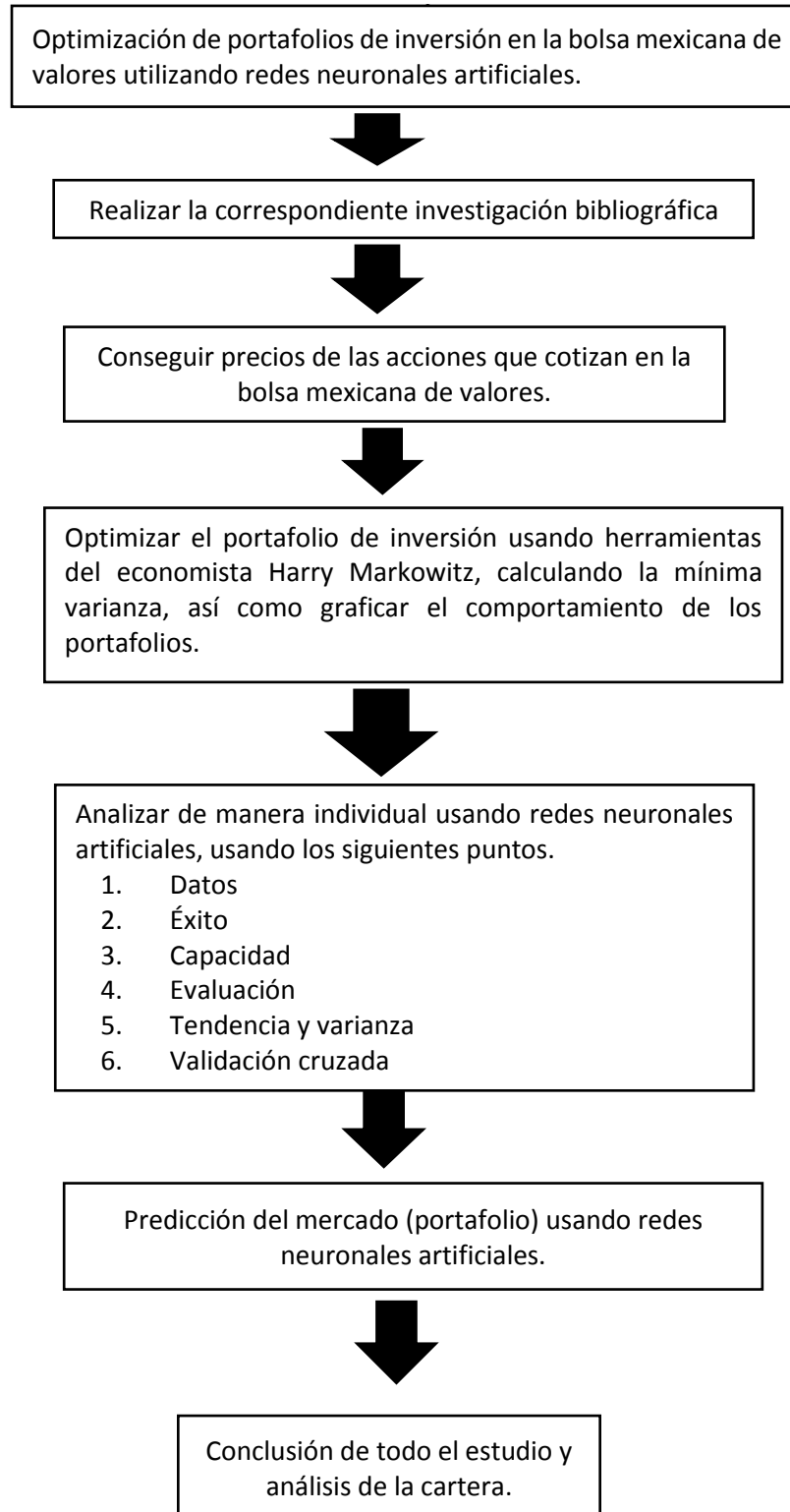
Sujeto a; Ec.34 y 35.

$$\mathbf{1} = \mathbf{w}^T \mathbf{1}_n \quad \text{Ec (34)}$$

$$\mathbf{w} > \mathbf{0}_n \quad \text{Ec (35)}$$

[11] [12].

Metodología.



Asumiendo que el inversionista quiso usar la teoría de portafolios para crear una cartera de inversión compuesto por Elektra, Bimbo, Amazon, BanBajío y Oro, para esto se extrajo la información histórica de los activos ya mencionados desde el 2016 hasta a inicios del mes de enero del 2022. Se hizo un registro de los datos históricos en Excel de la página *investing.com* desde el año 2016 hasta finales del año 2021.

Optimización del portafolio de inversión de H. Markowitz. (Teoría de portafolios)

En el algoritmo se introdujeron los símbolos de los activos para identificar las columnas a revisar, posteriormente se calculó los retornos logarítmicos para todas las series temporales, a su vez se trazó la serie de tiempo financiera normalizada para los símbolos seleccionados, después; se pasó a calcular el retorno, la volatilidad y el Sharpe ratio.

En este caso, estudiando las combinaciones de riesgo y retorno, usando la solución de simulación de Monte Carlo para aleatorizar los pesos, estos pesos suman un cien por ciento, Dentro del código; se simularon los pesos de cartera que suman el 100%, se calculó el retorno y volatilidad, así como el resultado del Sharpe ratios. Después se hizo una prueba de retrospección desde el año 2016, el código derivó los pesos del portafolio para cada año que maximizó el Sharpe ratio, con esto se especificaron los límites para los pesos de activos individuales, al igual que todos los pesos deben sumar 100%. Así que, se hizo un conjunto de datos relevantes para el año dado, extrayendo las ponderaciones de la cartera que maximizó el índice de Sharpe, dando almacenamiento a estos pesos en un *dict object*.

Después; realizando una retrospección, el código comparó las estadísticas de cartera esperadas (a partir de las composiciones óptimas del año anterior aplicadas a los datos del año previo) con las estadísticas de cartera realizadas para el año actual (a partir de la composición óptima del año anterior aplicada a los datos del año actual), que son; las estadísticas de portafolios esperados y las estadísticas de cartera realizadas.

Lo siguiente, comparando las volatilidades de cartera esperadas y realizadas para los años individuales, la teoría del portafolio (MVP) hace un buen trabajo al predecir la volatilidad de la cartera, esto también está respaldado por una correlación relativamente alta entre las dos series temporales. Después, se comparan los rendimientos de cartera esperados con los realizados y, por último; se hace una comparación del índice de Sharpe, lo esperado con lo realizado.

La siguiente parte, es acerca del *Machine Learning*, para la serie financiera de cada activo de manera individual. Se realizaron diferentes tareas (estimación o clasificación). Después, una comparación de rendimiento, como el (MSE) o la conexión de precisión. Luego está el aprendizaje medido por la mejora en el desempeño dada la experiencia del algoritmo con la tarea. La clase de tareas se describió en general, en función del conjunto de datos dados, incluyendo los datos de características y los datos de etiquetas en el caso de aprendizaje supervisado, o solo la información de características en el caso de ANS.

Usar RNA para analizar un portafolio de inversión

Analizar de manera individual usando RN, con los siguientes puntos.

Información o Datos.

Estos datos de muestra se crearon con base a una línea de intervalo de tiempo real financiero para el perfil intercambio del activo. Primero, los datos se importaron desde un archivo CVS y luego se volvieron a muestrear datos mensuales y se almacenaron en un *Series object*.

Para tener una sola característica, el código creó un vector de características sintéticas. Permitiendo visualizaciones simples en dos dimensiones. Las características sintéticas (variable independiente), (etiquetas de datos, variable dependiente). En lo que sigue, también se abstrajo del hecho de que los datos de las etiquetas son de naturaleza secuencial y temporal.

El monto de información como ejemplo, trató este modelo como un monto de valores generales compuesto por vectores de características unidimensional y un vector de etiquetas unidimensional.

Para esto se transformaron los datos de las etiquetas a un objeto *ndarray*, del cual se tomó el valor medio de los datos de elemento a elemento, creando una característica sintética como un objeto *ndarray*.

Éxito.

La medida del éxito para los problemas de estimación en general es el MSE, con base en al MSE; el éxito se definió con los datos de las etiquetas como el punto de referencia relevante y los valores pronosticados de un algoritmo después de haber estado expuesto al conjunto de

información considerando dos algoritmos de regresión lineal (OLS) y el de redes neuronales.

Primero se hizo la regresión OLS; después; la función MSE calculó el error cuadrático medio, así como haciendo el ajuste del modelo de regresión OLS, e incluyendo monomios de quinto orden, con esto; haciendo la predicción del modelo de regresión OLS dados los parámetros óptimos y como siguiente paso calculando el valor del MSE dados los valores de la predicción.

Implementando el paso de regresión basado en los subconjuntos más pequeños, mientras que los pasos de predicción se implementan en función del monto de información de características por cada situación. En general, el MSE cae significativamente cuando se aumenta la relación de información de adiestramiento; para esto se hace el paso de regresión basado en el subconjunto de datos, después; la predicción basada en el conjunto de datos completo y se obtiene el valor MSE resultante.

En segundo lugar, está la red neuronal; haciendo aproximación de los datos de la muestra, la red neuronal solo tiene una capa oculta. Realizando el paso de ajuste con un número relativamente alto de épocas, después el paso de predicción que también aplana el objeto *ndarray*, siguiente paso; fue el valor MSE para la predicción de la DNN.

Con el paquete Keras, los valores MSE se almacenaron después de cada paso de aprendizaje.

Capacidad

La capacidad de un modelo o algoritmo define qué tipos de funciones o relaciones del modelo o algoritmo básicamente puede aprender. En el caso de la regresión OLS basada en solo monomios, solo hay un parámetro que define la capacidad del modelo; el grado del monomio más alto a utilizar. Si este parámetro de grado se establece en $\text{grado} = 3$, el modelo de regresión OLS puede aprender relaciones funcionales de forma constante, tipo lineal; cuadrático o cúbico. Cuanto mayor sea el grado del parámetro, mayor será la capacidad del modelo de regresión OLS. Empezando en $\text{grado} = 1$ y aumentando el grado en incrementos de dos en dos.

En el paso de regresión para diferentes valores de *deg*, la capacidad de una red neuronal depende de una serie de hiperparámetros, que son;

- Cantidad de superficies ocultas
- cantidad de unidades ocultas para cada superficie oculta

Juntos, estos dos hiperparámetros definieron el número de parámetros entrenables (pesos) en la RN. El ejemplo de RN tiene un número cambiante bajo de parámetros entrenables. Añadir sólo una capa más del mismo tamaño, aumenta el número de parámetros entrenables significativamente.

Parte de esto, se agregó potencialmente una variedad capas a la red neuronal, una RNA profunda con 3 zonas escondidas.

Evaluación

El análisis es centrado en la capacidad de los códigos de estimación sobre el grupo de información de la muestra como un todo. Como regla general, la capacidad del modelo o algoritmo influye directamente en su rendimiento al entrenarlo y evaluarlo en el mismo conjunto de datos. Sin embargo, este es el "caso simple y fácil" en el ML. El caso complejo es cuando se usa un modelo o algoritmo entrenado para una generalización sobre datos que el modelo o algoritmo no ha visto antes. Tal generalización es, la predicción (estimación) del precio futuro de una acción, dada la historia de los precios de las acciones, aunque el término predicción a menudo se usa libremente en el contexto de las estimaciones dada la cantidad de información de características usándolo para el acondicionamiento, una predicción real, probablemente implica predecir algo no conocido de antemano. En general, la información se reparte en subconjuntos que tienen diferentes propósitos; como los siguientes:

Cantidad de información de entrenamiento

Este es el subconjunto utilizado para acondicionar el código.

Cantidad de información de verificación

Este subconjunto se utiliza para validar el rendimiento del algoritmo durante el acondicionamiento y este monto de información es diferente del monto de valores del acondicionamiento

Cantidad de información de prueba

Este fue el subconjunto en el que el algoritmo entrenado solo se prueba después del entrenamiento; una vez terminado este.

La información que se tiene al aplicar un algoritmo entrenado (actualmente) en la verificación de la información podría reflejarse en el entrenamiento en sí mismo (por ejemplo, ajustando los hiperparámetros de un modelo). Por otro lado, la idea es que las percepciones de probar el algoritmo entrenado con la información de prueba no se reflejaron

en el propio entrenamiento ni en los hiperparámetros. En lo siguiente se elige, de forma arbitraria, el 25 % de los datos de muestra para pruebas; el modelo o algoritmo no ve estos datos antes de que se completara el entrenamiento (aprendizaje) terminado. De manera similar, el 25% de los datos de la muestra se reserva para validación; estos datos son usados para monitorear el desempeño durante el paso de entrenamiento y posiblemente durante el aprendizaje. El 50% restante se utiliza para la formación (aprendizaje) propiamente dicha.

Dado el conjunto de datos de muestra, tiene sentido aplicar técnicas de aleatorización para poblar todas las muestras de subconjuntos para datos al azar, con esto se define la cantidad de referencias de un monto de información para la prueba, número de referencias del grupo de información de validación; índice aleatorio para la cantidad de información completa, índices ordenados para las sub cantidades de valores; resultados para los subconjuntos de valores de características y subconjuntos de datos de etiquetas resultantes.

Muestreo aleatorio.

La población aleatoria de la cantidad de información para el adiestramiento, verificación y examen son una técnica común y útil para conjuntos de datos que no son secuenciales ni de naturaleza temporal. Sin embargo, cuando se trata; por ejemplo, de una serie de tiempo financiera, generalmente se debe evitar mezclar los datos porque rompe las estructuras temporales e introduce un sesgo de previsión en el proceso al usar, por ejemplo, muestras posteriores para el entrenamiento y la implementación de las pruebas en muestras anteriores.

En función de los subconjuntos de datos de capacitación y validación, se implementa una regresión para diferentes valores de parámetros de grados y se calcula los valores de MSE para las predicciones en ambos subconjuntos de datos. Aunque los valores del MSE en el conjunto de valores de adiestramiento disminuyen monótonamente, estos valores del MSE en la cantidad de información de verificación a menudo alcanzan el mínimo con un determinado valor de parámetro y luego aumentan nuevamente. Este fenómeno indica lo que se denomina sobreajuste. Se hacen los ajustes de regresión para los diferentes valores de grados y se comparan los ajustes para la cantidad de información de adiestramiento y validación. Se calcula el valor del MSE para la cantidad de valores de enseñanza, así como los datos del MSE para la información de validación.

Con Keras y el modelo con la red neuronal, este rendimiento de información de validación

se puede monitorear para cada paso de aprendizaje. También se puede usar funciones de devolución de llamada para detener el entrenamiento del modelo antes de tiempo cuando no se observan más mejoras, por ejemplo, en el rendimiento de información para enseñanza, se hace uso con dicha función de devolución de llamada, se grafican las predicciones de la red neuronal para la información de enseñanza, validación, datos de entrenamiento y validación incluyendo ajustes de regresión.

El aprendizaje se detiene en función del valor MSE de los datos de entrenamiento, y sólo se detiene después de un cierto número de épocas que no muestran una mejora; los mejores pesos se restauran cuando se detiene el aprendizaje y se especifican los subconjuntos de datos de validación. La función *callback* (llamado) se pasa al método *fit()*.

Keras permite el análisis del cambio en los valores de MSE en ambos conjuntos de datos para cada época en la que se entrena el modelo, se grafican los valores de MSE donde disminuye con el número creciente de épocas de entrenamiento, aunque solo en promedio y no de forma monótona:

En el caso de la regresión OLS, se elige un valor alto, pero no tanto, para el parámetro de grado, como el grado=9. La parametrización del modelo de red neuronal proporciona automáticamente la mejor configuración del modelo al final del entrenamiento. Se comparan las predicciones de ambos modelos entre sí y con la cantidad de números de prueba.

Sesgo y varianza.

Un problema en el ML en general y cuando se aplican algoritmos de ML a datos financieros en particular es el problema del sobreajuste. Un modelo sobre ajusta sus datos de entrenamiento cuando el rendimiento es peor en la información de validación y enseñanza que en los valores de adiestramiento. En el siguiente paso se utilizan subconjuntos más pequeños tanto para el entrenamiento como para la validación e implementa una regresión lineal, así como una de orden superior. El ajuste de regresión lineal tiene un alto sesgo en la cantidad de información de enseñanza; Las restas absolutas sobre las predicciones y los datos de las etiquetas son relativamente altas. Para esto se tiene un subconjunto de datos de características más pequeñas, un subconjunto de datos de etiquetas igual más pequeñas; se realiza la regresión OLS de alto sesgo (lineal), así como la regresión OLS de alta varianza (alto orden) y un subconjunto de datos de características para graficar.

Se calculan no solo los valores de MSE, sino también los valores de R2, también se hacen los ajustes de regresión OLS de alto sesgo y alta varianza, mostrando el sesgo del modelo como diferencias medias absolutas; se calcula la varianza del modelo como varianza de las predicciones del modelo, el rendimiento del modelo de alto sesgo en la información de enseñanza, ganancia del ejemplo de alto sesgo para los datos de validación, rendimiento del modelo de alta varianza en la información de adiestramiento y ganancia del ejemplo de alta varianza en los datos de validación.

Validación cruzada.

Un enfoque estándar para evitar el sobreajuste es la validación cruzada, durante la cual se prueban múltiples poblaciones de datos de entrenamiento y validación. El paquete Scikit-Learn proporciona la funcionalidad para implementar la validación cruzada de forma estandarizada. La función *cross_val_score* puede aplicar a cualquier objeto modelo de scikit-learn, así; se implementa el enfoque de regresión OLS en el conjunto de datos de muestra completo, utilizando un modelo de regresión OLS polinomial de scikit-learn. La validación cruzada quíntuple se aplica para diferentes grados para el polinomio más alto. Las puntuaciones de la validación cruzada se vuelven peores, en promedio; cuanto mayor sea el grado más alto en la regresión. De manera similar, los mejores puntajes de validación se observan para el 20% medio del conjunto de datos de muestra:

Validación cruzada, se crea una clase de modelo de regresión polinómica, se ajusta la configuración de impresión por defecto para *numpy*, se implementa la validación cruzada de cinco desgloses, Keras proporciona clases de almacenamiento para usar objetos de modelo con la funcionalidad scikit-learn, como la función *cross_val_score*, utilizando el Clase *KerasRegressor* para envolver los modelos de redes neuronales y aplicarles la validación cruzada. Las puntuaciones de validación cruzada son mejores en comparación con las puntuaciones de validación cruzada de la regresión OLS, la capacidad de la red neuronal no juega un papel importante en este tema.

Los siguientes pasos como la; clase de envoltura para red neuronal con baja capacidad, validación cruzada para red neuronal con baja capacidad, Clase *Wrapper* para redes neuronales de alta capacidad, validación cruzada para redes neuronales de alta capacidad, son pasos primordiales realizados [13] [14] [15] [16] [17] [18].

Predicción del mercado (portafolio) usando redes neuronales artificiales.

Esta parte combina las finanzas basadas en datos con el enfoque de aprendizaje automático del procedimiento anterior. Las redes neuronales se utilizan para descubrir ineficiencias estadísticas en los “Mercados eficientes”, analizando la teoría del ME y usando la regresión de OLS para ilustrar con base a los datos de series temporales financieras. “Predicción del mercado basada en datos de rendimiento”, aplicando redes neuronales, junto con la regresión OLS, para adivinar la dirección adelantada del valor de una herramienta financiera (“dirección del mercado”). El análisis se basa únicamente en datos de retorno. “Predicción de mercado con más funciones”, agregando más funciones a la combinación, como indicadores financieros típicos. En este tema, los primeros resultados indican que las ineficiencias estadísticas podrían estar presentes. Esto se confirma en “Predicción del mercado intradía (dentro del día)”, que funciona con datos intradía en comparación con los datos al final del día.

Primero, importe los datos y su visualización, luego se lee los datos en un objeto *DataFrame*, después; planificando los datos de la serie temporal normalizada, en segundo lugar, los datos de precios de todas las series temporales financieras se retrasan y se almacenan en objetos *DataFrame*, el número de rezagos (en días de negociación), siguiente, creando un nombre de columna, después retrasando los datos de los precios, se agrega el nombre de la columna a un objeto de lista, se borran todas las filas de datos incompletos, se crean los datos rezagados para cada serie de tiempos financieros.

Almacena los resultados en un objeto *dict*, mostrando una muestra de los datos de precios retardados. En tercer lugar, con los datos preparados, el análisis de regresión OLS es sencillo de realizar mostrando los resultados de regresión óptimos promedio. Sin duda, el dato de precios con un desfase de solo un día tiene el mayor poder explicativo, su peso es cercano a 1, lo que respalda la idea de que el mejor predictor del valor para mañana de una herramienta financiera es su valor de hoy. Esto también es válido para los resultados de regresión simple obtenidos por serie temporal financiera:

Para esto se obtienen los datos de la serie temporal actual, se implementa el análisis de regresión, se almacenan los parámetros óptimos de regresión en un objeto *dict*, combinando

los resultados óptimos en un único objeto *ndarray*, se ponen los resultados en un objeto *DataFrame* y los muestra, visualizando el promedio de los parámetros óptimos de regresión (pesos) para cada *lag*.

Otro enfoque importante es; la estacionariedad de los datos de la serie temporal, que también se comprueba mediante los siguientes pasos: Mostrando la correlación entre las series temporales retardadas, después; haciendo el examen de fijación mediante el *Dickey Fuller*

Predicción de mercado basada en datos de rentabilidad.

Con este fin, se utilizan los mismos datos recopilados de la BMV para obtener rendimientos logarítmicos a partir de los precios. La idea es medir la capacidad de regresión OLS con la capacidad de las redes neuronales para predecir la dirección del movimiento del día siguiente para las diferentes series temporales. El objetivo en esta etapa es descubrir las ineficiencias estadísticas en comparación con las ineficiencias económicas. Lo primero en el estudio es creando conjuntos de datos con información de retorno de registros retrasados. También se prueba la estacionariedad de los datos de retorno de registro retardados normalizados, y las características se prueban para determinar la correlación.

Teniendo el logaritmo de los rendimientos a partir de los datos de los precios, junto con los siguientes que son: atrasando los datos de los retornos logarítmicos, aplicando la normalización gaussiana a los datos de las características, así como; mostrando los datos de los retornos rezagados, probando la estacionariedad de los datos de la serie temporal, y mostrando los datos de correlación de las características, es una manera de tener un análisis mejor hecho en este tema.

Después, primer punto; se implementa la regresión OLS y se generan las predicciones resultantes de la regresión. El estudio se aplica con el conjunto de datos completo, así como mostrando qué tan bien funcionan los algoritmos en la muestra. La precisión con la que la regresión OLS predice la dirección del movimiento del día siguiente es ligeramente superior al 50 %, o incluso unos pocos puntos porcentuales, con algunas excepciones, que son: el paso de regresión, la etapa de predicción y la precisión de la predicción.

En segundo lugar, se vuelve a realizar el mismo análisis, pero esta vez con una red

neuronal de Scikit-Learn como modelo para el aprendizaje y la predicción. La precisión de la predicción en la muestra es significativamente superior al 50 % en todo momento y superior al 60 % en algunos casos, parte del proceso es la instanciación del modelo, ajuste del modelo, la etapa de predicción y el cálculo de la exactitud.

Tercero, el mismo análisis nuevamente, pero con una red neuronal del paquete Keras usando el *backend Tensorflow*, así como la función de creación de modelos, instanciación o que es lo mismo; la especificación de información del modelo, ajuste del modelo, etapa de predicción y cálculo de la precisión

Estos pasos muestran que las redes neuronales pueden superar significativamente la regresión OLS en la muestra para predecir la dirección de los movimientos de precios del día siguiente. Sin embargo, ¿cómo cambia la imagen cuando se prueba el rendimiento fuera de la muestra de los dos tipos de modelos? Con este fin, se repiten los análisis, pero el paso de entrenamiento (ajuste) se implementa con el 80 % de los datos mientras que el rendimiento se prueba en el 20 % restante. La regresión OLS se implementa primero. La regresión OLS fuera de la muestra, enseña los niveles de precisión similares a los de la muestra, alrededor del 50 %, para esto se crea el subconjunto de información de entrenamiento, también se crea el subconjunto información de prueba.

Eficiencia del mercado de forma débil.

Aunque la etiqueta como eficiencia de mercado de forma débil podría sugerir lo contrario, es la forma más difícil en el sentido de que solo se pueden usar datos relacionados con series de tiempo para identificar ineficiencias estadísticas. Con la forma semi fuerte, se podría agregar cualquier otra fuente de datos disponibles públicamente para aumentar la certeza de la predicción.

El simple análisis de patrones de retorno históricos basados en regresión OLS o redes neuronales puede no ser suficiente para descubrir ineficiencias estadísticas. Hay dos elementos principales del enfoque que se pueden ajustar con la esperanza de mejorar los resultados de la predicción:

Funciones: que además de los datos básicos de precios y rendimientos, se pueden agregar otras funciones a los datos; así como indicadores técnicos (por ejemplo, promedios móviles

simples o SMA para abreviar).

La esperanza es, en la tradición de los cartógrafos técnicos, que dichos indicadores mejoren la precisión de la predicción. Longitud de la barra, en lugar de trabajar con datos al final del día, los datos intradías pueden permitir mayores precisiones de predicción. Aquí, el punto es que; sea más probable que se descubran ineficiencias estadísticas durante el día en comparación con el final del día, cuando todos los participantes del mercado en general prestan la mayor atención a realizar sus operaciones finales teniendo en cuenta toda la información disponible.

Predicción de mercado con más funciones.

En el comercio, existe una larga tradición de utilizar indicadores técnicos para generar, en función de los patrones observados, señales de compra o venta. Dichos indicadores técnicos, básicamente de cualquier tipo, también pueden utilizarse como características para el entrenamiento de redes neuronales, para esto se hicieron y aplicaron los siguientes puntos: una SMA, valores máximos y mínimos móviles, impulso y volatilidad móvil como características: Media móvil simple SMA, Mínimo móvil, Máximo móvil, Momento como media de los rendimientos logarítmicos, volatilidad y dirección como característica binaria.

Indicadores técnicos como funciones.

En la muestra, el rendimiento del modelo MLPClassifier ahora es mucho mejor cuando se tienen en cuenta las nuevas funciones y se normalizan para el entrenamiento. El modelo Secuencial de Keras alcanza precisiones en torno al 70% para el número de épocas entrenadas. Por experiencia, estos pueden incrementarse fácilmente aumentando el número de épocas y/o el soporte de la RN y normalizar los datos de las condiciones.

Lo siguiente, haciendo el análisis, esta vez con la división de entrenamiento y prueba como se usó antes. Creando una imagen mixta en el mejor de los casos. Los números no representan mejoras reales en comparación con el enfoque, ya que solo se basan en datos de retorno retrasados como características. Para instrumentos seleccionados, parece haber una ventaja de algunos puntos porcentuales en la precisión de la predicción en comparación con el punto de referencia del 50 %. Para otros, sin embargo, la precisión aún está por debajo del 50 %,

como se ilustra para el modelo *MLPClassifier*, los valores estadísticos de la cantidad de enseñanza se utilizan para la estandarización.

El buen rendimiento para la muestra y el rendimiento no tan bueno fuera de esta, sugiere que el sobreajuste de la red neuronal podría desempeñar un papel crucial. Un enfoque para evitar el sobreajuste es utilizar métodos de conjunto que combinen múltiples modelos entrenados del mismo tipo para generar un metamodelo más sólido y mejores predicciones fuera de la muestra. Uno de estos métodos se llama embolsado. Scikit-learn tiene una implementación de este enfoque en forma de la clase *BaggingClassifier*. El uso de múltiples estimadores permite entrenar a cada uno de ellos sin exponerlos a la cantidad de información de adiestramiento completo o a todas las características. Esto debería ayudar a evitar el sobreajuste. El siguiente paso es, implementando un enfoque de embolsado basado en varios estimadores base del mismo tipo (*MLPClassifier*). En general, el embolsado parece evitar, al menos hasta cierto punto, el sobreajuste y mejora notablemente las predicciones; los puntos estratégicos son: el estimador base, el número de estimadores utilizados, porcentaje máximo de datos de entrenamiento utilizados por estimador y número máximo de características utilizadas por estimador [19] [20] [21] [22] [23] [24] [25] [26].

Resultados y discusión.

Teoría del portafolio óptimo

Se seleccionaron cinco empresas diferentes incluyendo el oro las cuales son, Elektra, Bimbo, Amazon, BanBajío; con esto se vio el comportamiento de todo el conjunto. Los datos se consiguieron en la página *investing.com*, la información se colocó en un archivo CSV y este se usó para el código. El programa Python 3.8.8 nos ayudó a calcular el retorno de la serie temporal de la información, para eso se introdujeron los símbolos de cada acción para identificarlos véase la figura 1. La Figura 2 muestra un comparativo de los precios de cada símbolo seleccionado desde el año 2016 hasta el 2022.

```

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1420 entries, 2016-05-27 to 2022-01-14
Data columns (total 22 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   ELEKTRA     1420 non-null   float64
1   BIMBOA     1420 non-null   float64
2   AMZN       1420 non-null   float64
3   BABAN      1420 non-null   float64
4   ORO        1420 non-null   float64
5   IPC        1420 non-null   float64
6   INMEX      1420 non-null   float64

```

Figura 1.

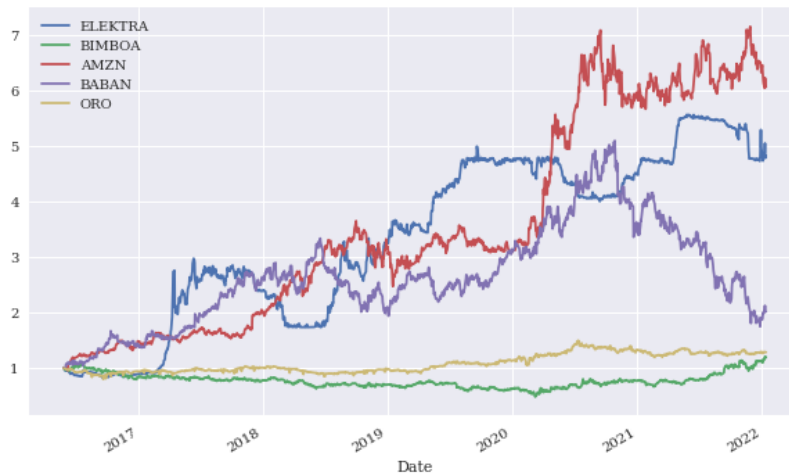


Figura 2.

Para esto se calculó el rendimiento, siendo de 0.16024182965083822 , la volatilidad de 0.1352829177067855 e índice de Sharpe de 1.1844941871977444 , asignando una serie de pesos para cada símbolo.

Se combinó el riesgo y rendimiento del portafolio junto con el de Sharpe, para esto; se aplicó la simulación Monte Carlo aleatorizando los pesos de la cartera (matriz de pesos), véase la figura 4. Con esto; se tuvo una suma del 100 por ciento, mostrándose el comportamiento de 1000 carteras. Véase la figura 3.

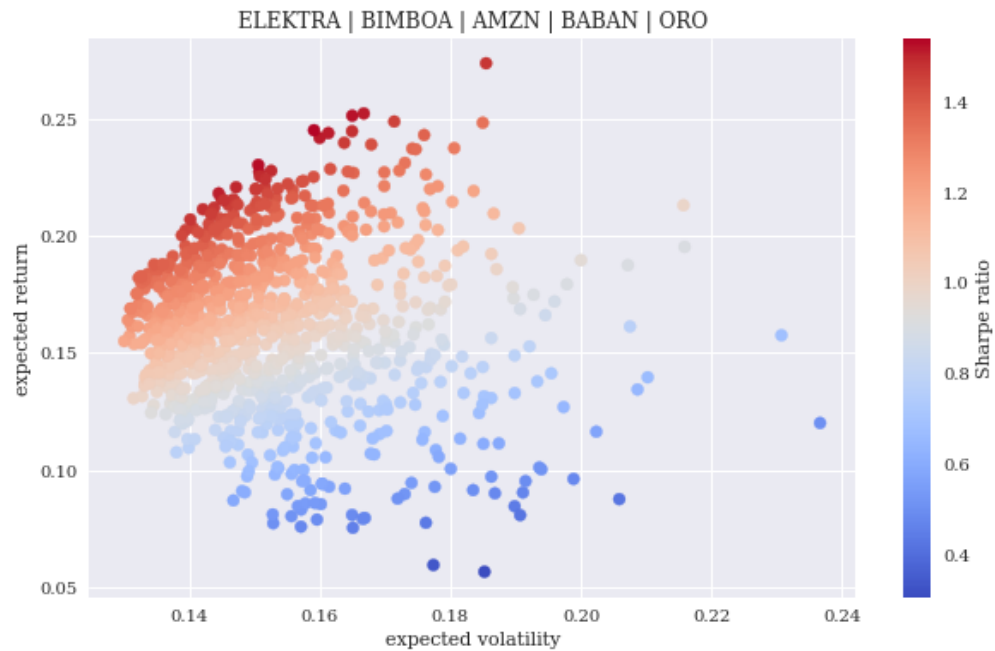


Figura 3. Volatilidad, rendimientos y ratios de Sharpe.

```
array([[ 0.004,  0.352,  0.263,  0.102,  0.279],
       [ 0.538,  0.136,  0.046,  0.004,  0.276],
       [ 0.254,  0.284,  0.286,  0.076,  0.100],
       [ 0.168,  0.069,  0.277,  0.291,  0.195],
       [ 0.191,  0.324,  0.284,  0.047,  0.154]])
```

Figura 4. Matriz de pesos.

También, se realizó un estudio del rendimiento de la cartera que se creó desde el 2016. Con esto se deriva las ponderaciones del portafolio para cada año significativo que eleva a tope el índice de Sharpe, véase figura 5.

```
{2016: array([ 0.000,  0.000,  0.686,  0.314,  0.000]),
 2017: array([ 0.085,  0.000,  0.457,  0.458,  0.000]),
 2018: array([ 0.737,  0.000,  0.207,  0.000,  0.056]),
 2019: array([ 0.486,  0.000,  0.000,  0.348,  0.166]),
 2020: array([ 0.000,  0.362,  0.203,  0.119,  0.316])}
```

Figura 5.

Se hizo una comparación de las estadísticas del portafolio esperado (a partir de la composición óptima del año anterior aplicada a los datos del año actual) con las estadísticas de cartera realizadas para el año actual (a partir de la composición óptima del año anterior aplicada a los datos del mismo año a datos del año actual)(epv = volatilidad del portafolio

esperado)(epr = retorno del portafolio esperado)(esr = Sharpe ratio esperado)(rpv = volatilidad del portafolio realizado)(rpr = retorno del portafolio realizado)(rsr = Sharpe ratio realizado). Véase figura 6 y 7.

	epv	epr	esr	rpv	rpr	rsr
2017	0.192325	0.646319	3.360552	0.196311	0.526412	2.681525
2018	0.180765	0.520394	2.878848	0.239685	-0.048738	-0.203341
2019	0.299239	0.830094	2.774016	0.104333	0.129460	1.240830
2020	0.123371	0.320882	2.600956	0.158237	0.047017	0.297130
2021	0.172650	0.200032	1.158593	0.141139	0.103358	0.732313

Figura 6.

```

epv    0.193670
epr    0.503544
esr    2.554593
rpv    0.167941
rpr    0.151502
rsr    0.949692
dtype: float64

```

Figura 7.

Después se comparó las volatilidades de los portafolios esperados y hechos para los años particulares. Esta teoría sirve para predecir la varianza de la cartera. También está respaldada por una correlación relativamente alta entre las dos series de tiempo. Véase figura 8 y 9.

	epv	rpv
epv	1.000000	-0.478856
rpv	-0.478856	1.000000

Figura 8.

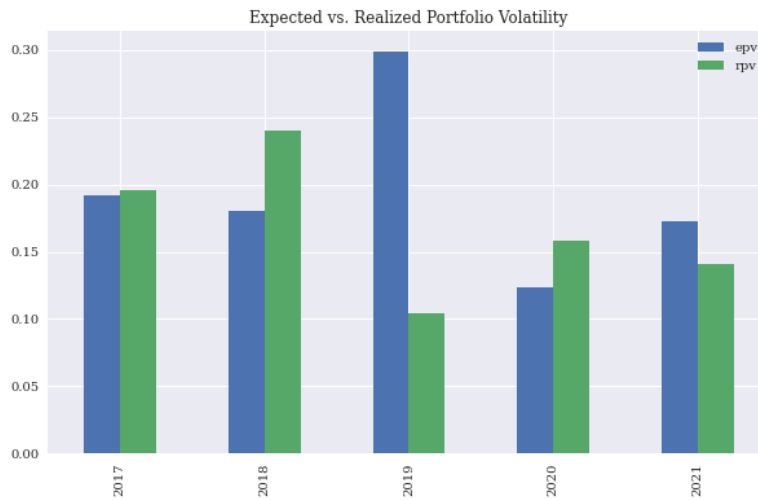


Figura 9.

Sin embargo, los resultados son opuestos cuando se comparan los rendimientos de cartera esperados con los realizados, figura 10. Ahora diciendo lo contrario, esta teoría falla al predecir los rendimientos del portafolio, como lo confirma la correlación negativa entre las dos series de tiempo. Véase figura 11.

	epr	rpr
epr	1.000000	0.345807
rpr	0.345807	1.000000

Figura 10.



Figura 11.

Los resultados son peores, con respecto al índice de Sharpe, figura 13. Para alguien que tiene como objetivo maximizar el índice de Sharpe del portafolio, las predicciones de la teoría están sumamente alejadas de los valores realizados. La correlación entre las dos series temporales es incluso menor que para los rendimientos. Véase figura 12.

	esr	rsr
esr	1.000000	0.370466
rsr	0.370466	1.000000

Figura 12.



Figura 13.

Análisis individual (por cada empresa) usando redes neuronales artificiales.

Datos

Se puso una sola característica, creando un vector de características sintéticas. Esto permitió la muestra del comportamiento de las empresas de manera gráfica; Elektra, Bimbo, Amazon, BanBajío y Oro. Véanse las figuras 14, 15, 16, 17 y 18.

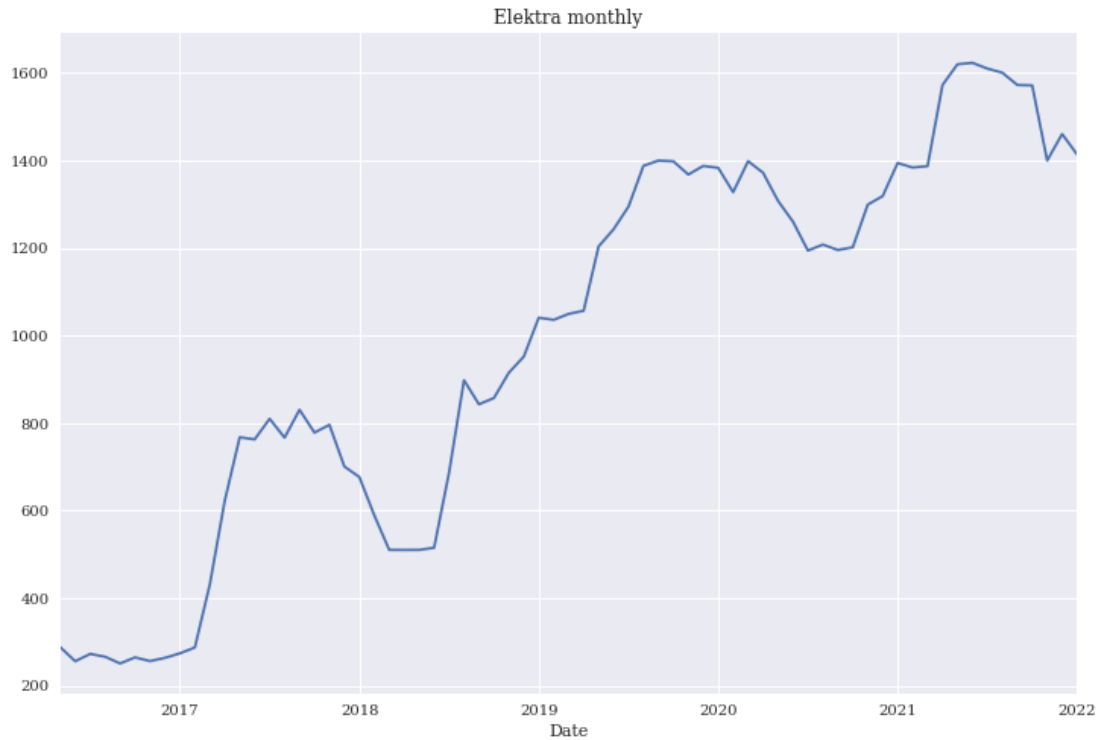


Figura 14. Variación mensual de Elektra.

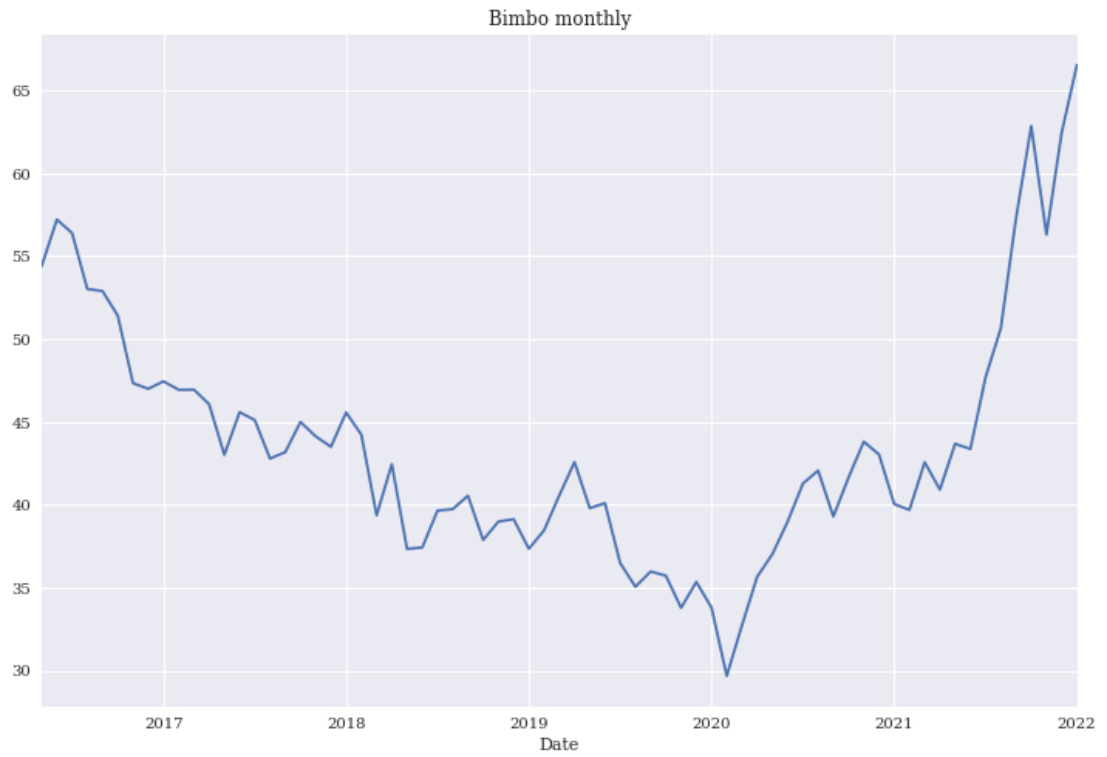


Figura 15. Variación mensual de Bimbo.

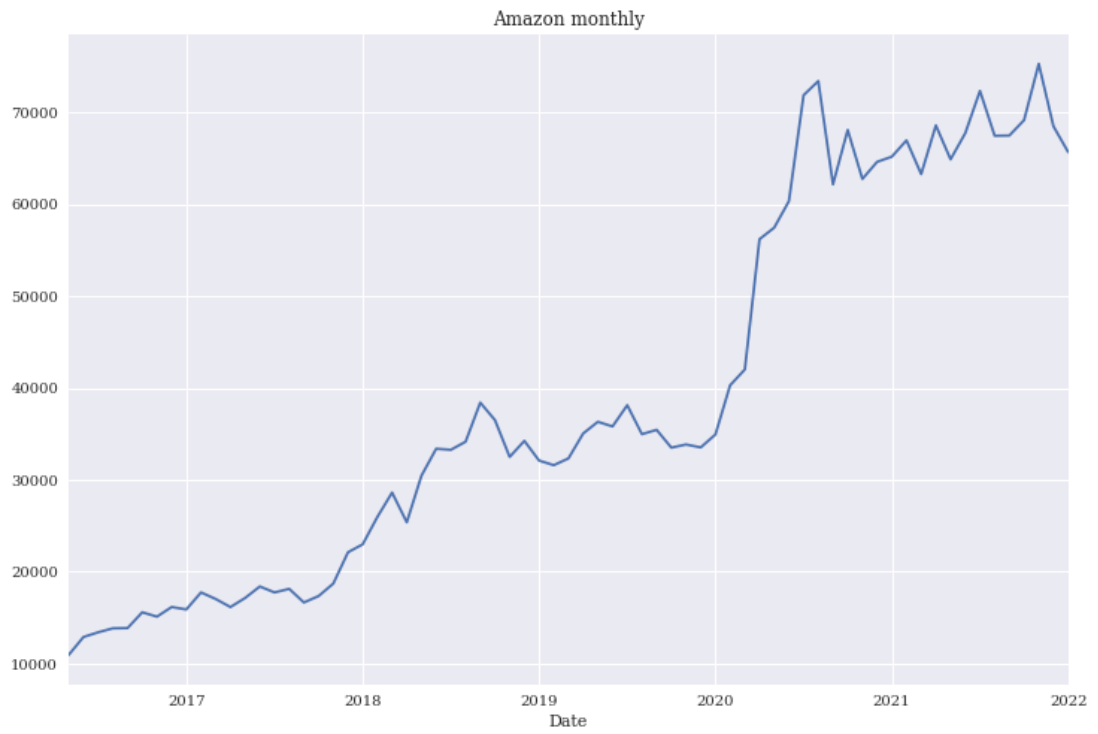


Figura 16. Variación mensual de Amazon.

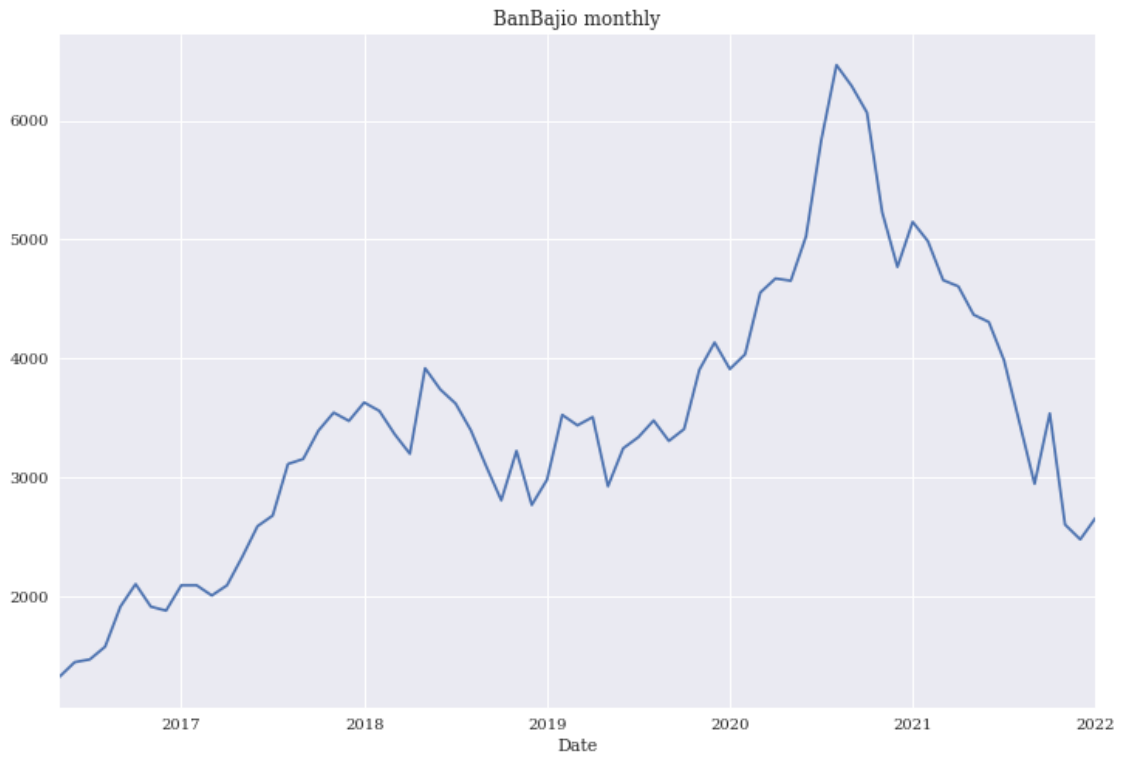


Figura 17. Variación mensual de BanBajo.



Figura 18. Variación mensual del Oro.

A continuación, se muestran los puntos sintéticos de la representación de las gráficas anteriores en el mismo orden. Véanse Las figuras 19, 20, 21, 22 y 23

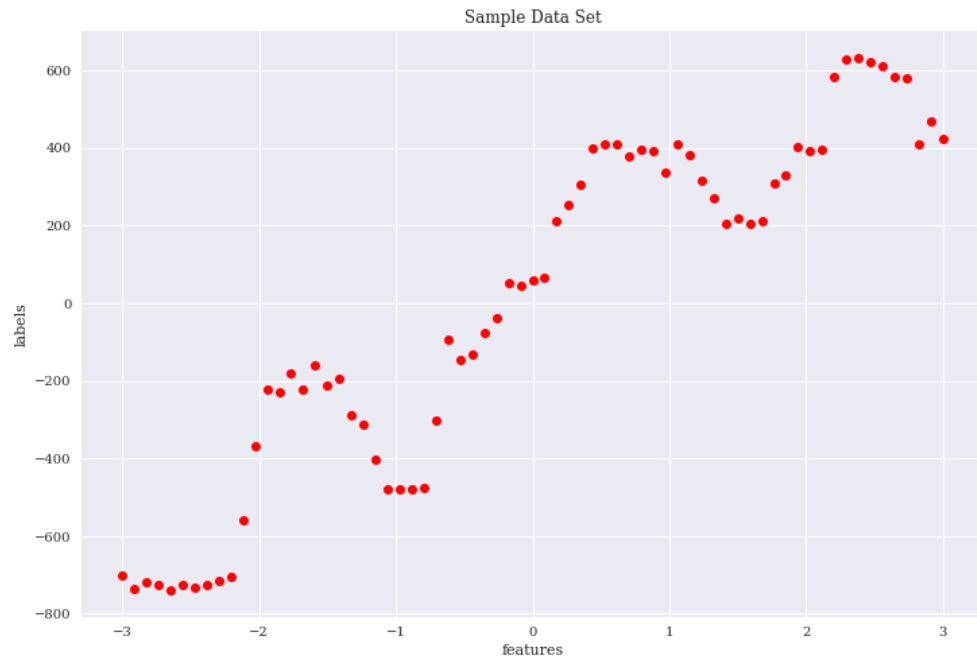


Figura 19. Puntos sintéticos de Elektra.

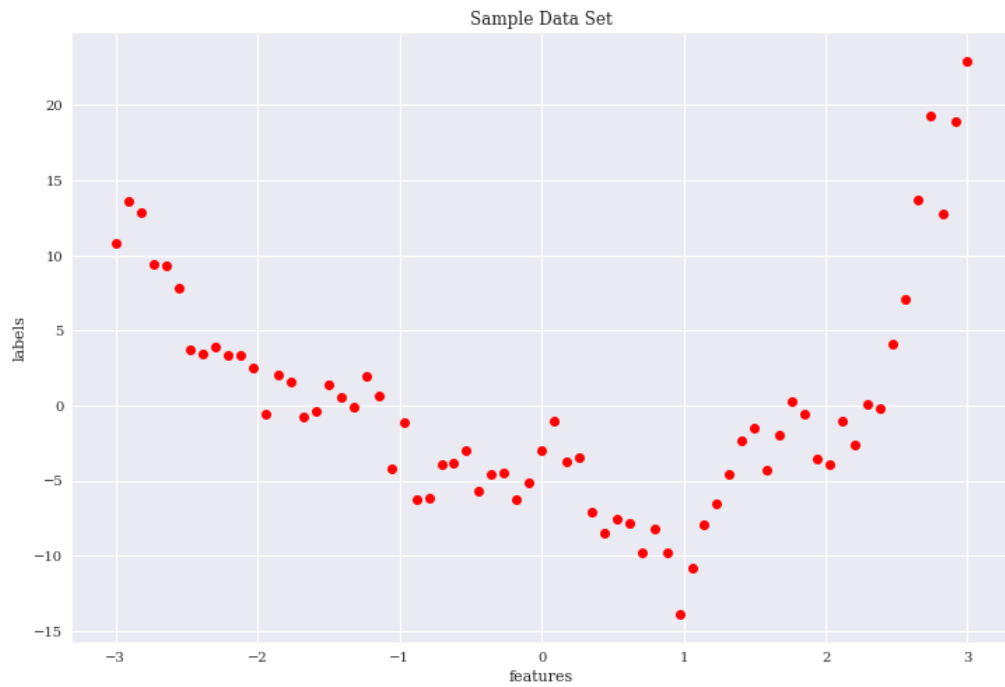


Figura 20. Puntos sintéticos de Bimbo.

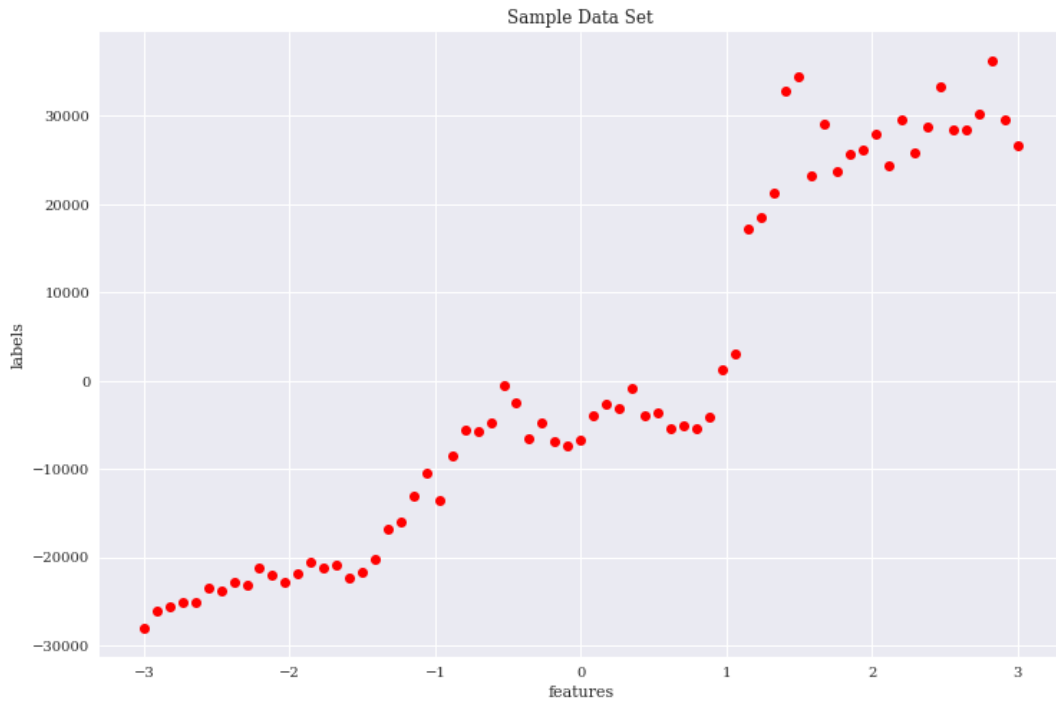


Figura 21. Puntos sintéticos de Amazon.

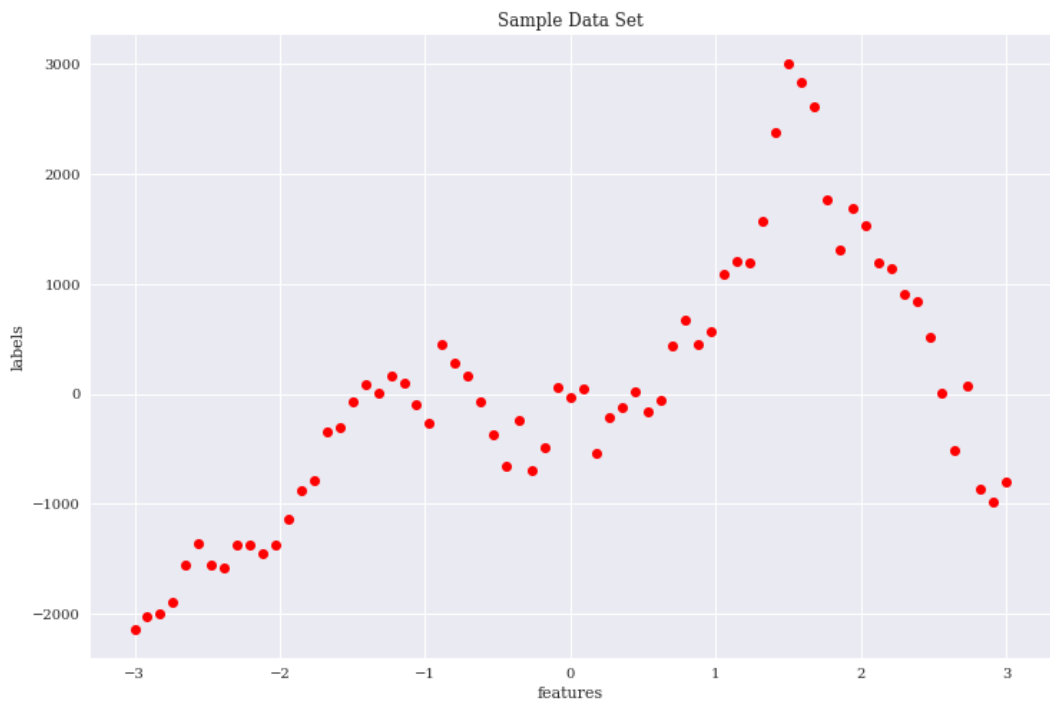


Figura 22. Puntos sintéticos de BanBajo.

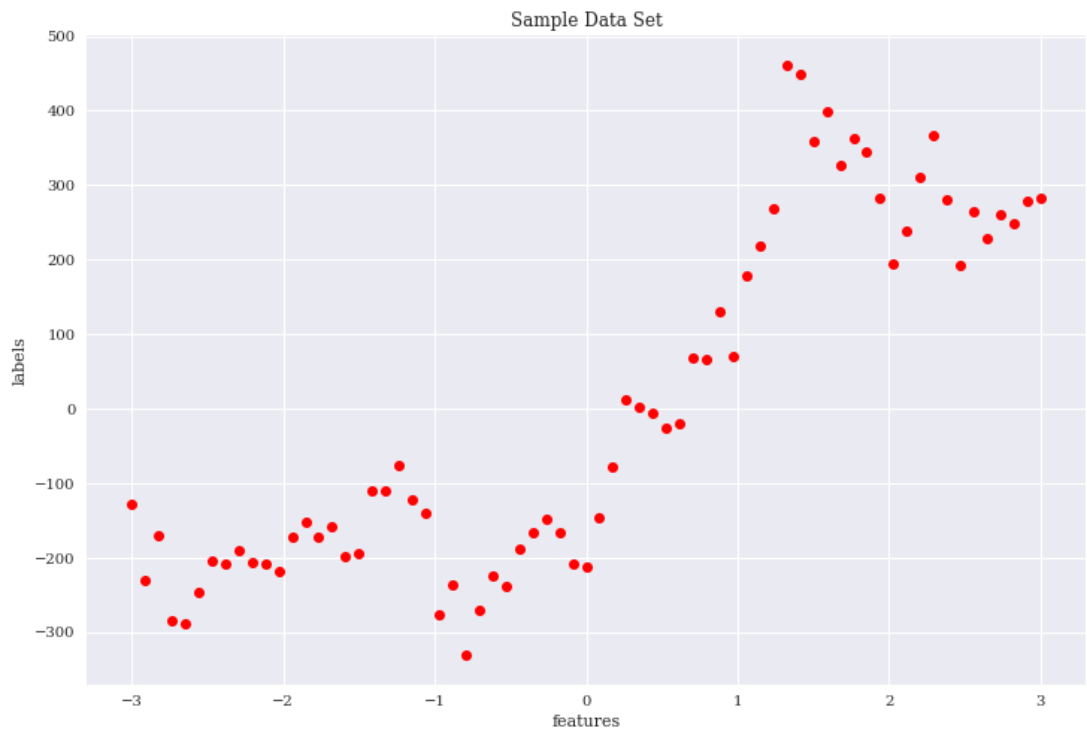


Figura 23. Puntos sintéticos del Oro.

Éxito

En la parte de éxito, como se explicó en el procedimiento. Se tuvo como resultado las siguientes graficas de la regresión. Véanse las gráficas como figuras 24, 25, 26, 27 y 28

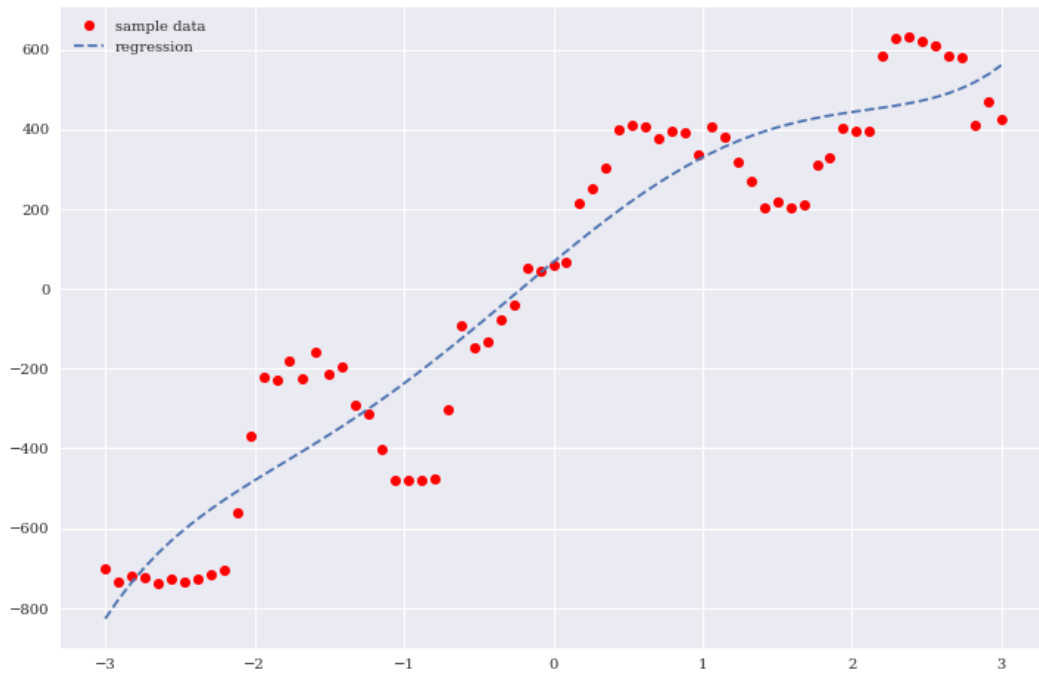


Figura 24. La predicción por el modelo de regresión OLS dados los parámetros óptimos de Elektra.

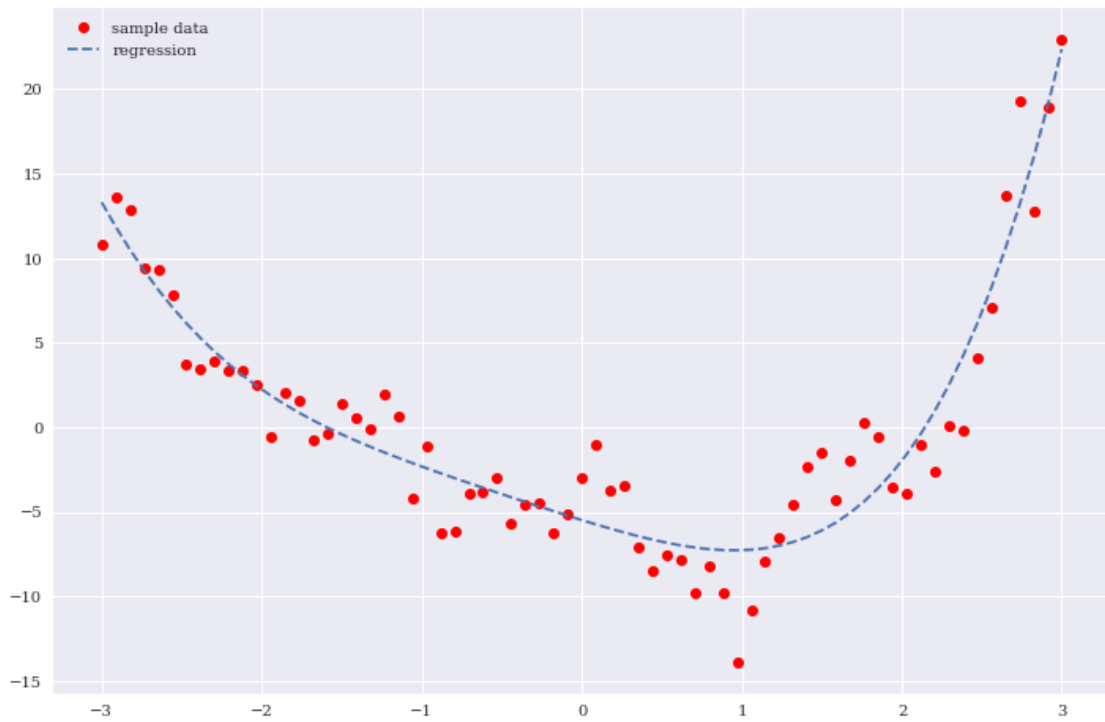


Figura 25. La predicción por el modelo de regresión OLS dados los parámetros óptimos de Bimbo.

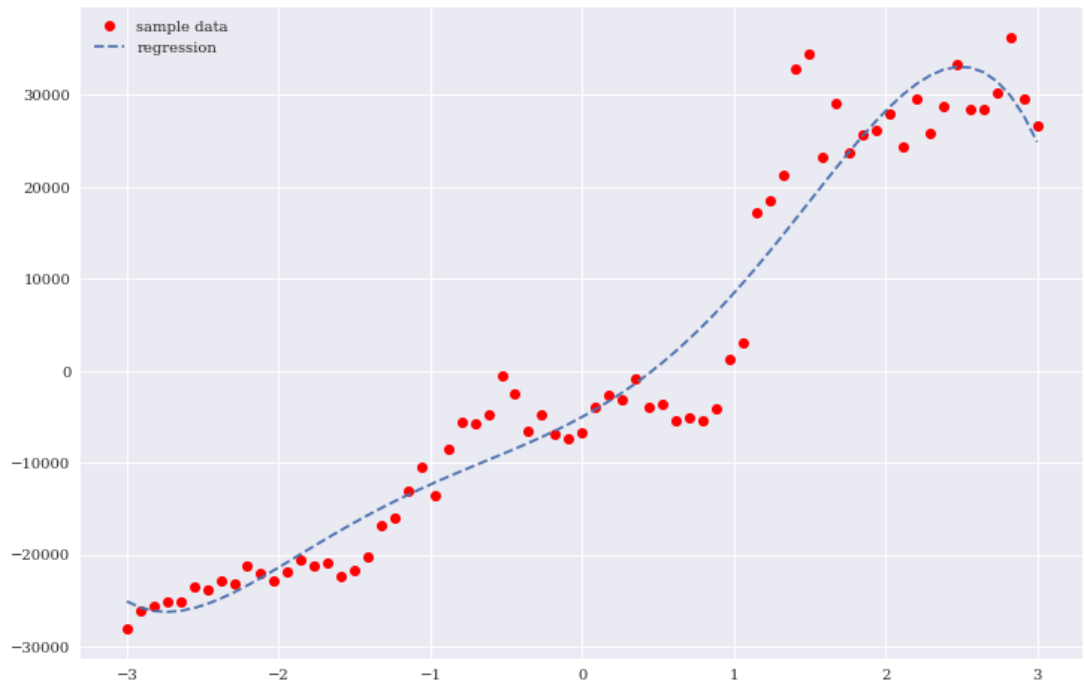


Figura 26. La predicción por el modelo de regresión OLS dados los parámetros óptimos de Amazon.



Figura 27. La predicción por el modelo de regresión OLS dados los parámetros óptimos de BanBajo.

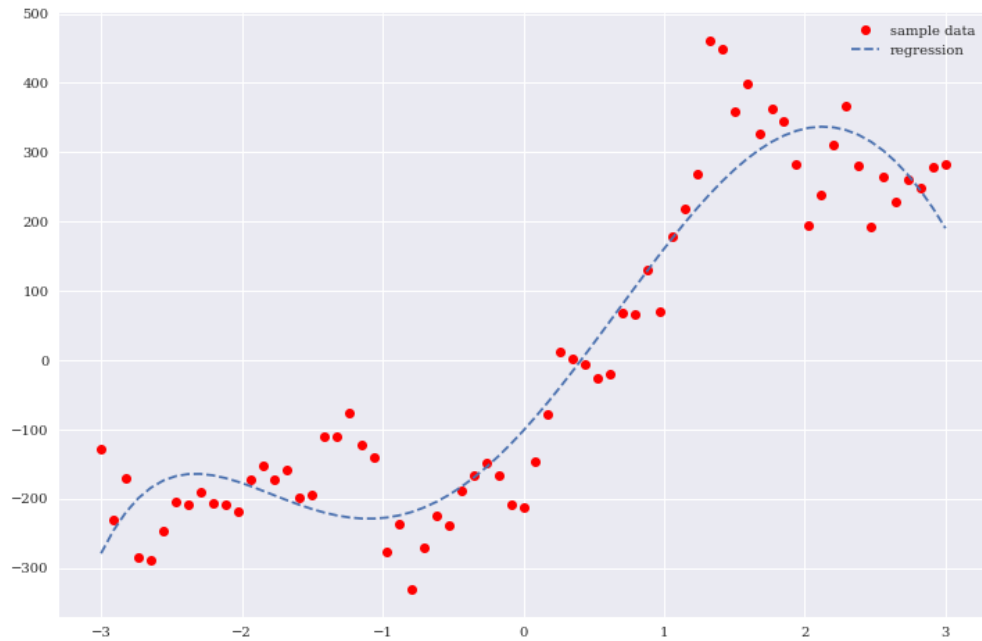


Figura 28. La predicción por el modelo de regresión OLS dados los parámetros óptimos del Oro.

En segundo lugar, se usó una red neuronal donde se muestra el modelo secuencial de cada red neuronal de cada activo, así como la gráfica en forma de figura de la aproximación de los datos. Vean se las figuras de los modelos secuenciales 29, 31; 33, 35 y 37; así como las gráficas en forma de figura 30, 32, 34, 36 y 38.

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	512
dense_1 (Dense)	(None, 1)	257

=====
 Total params: 769
 Trainable params: 769
 Non-trainable params: 0
 =====

Figura 29. Modelo secuencial de la red neuronal de Elektra.

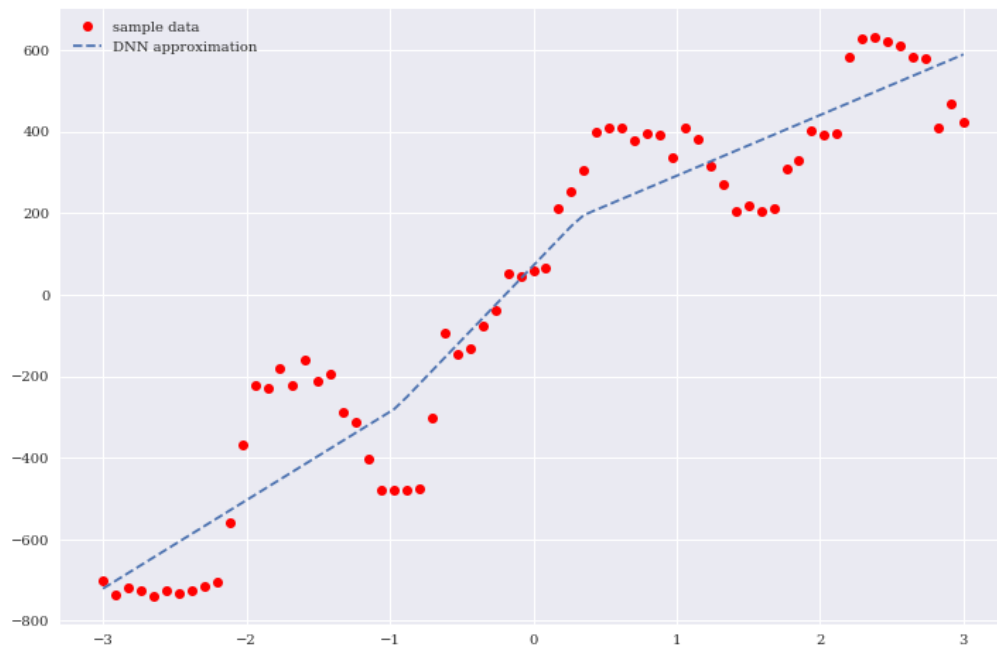


Figura 30. Aproximación de la muestra de la red neuronal profunda (DNN) de Elektra.

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	512
dense_1 (Dense)	(None, 1)	257

=====
 Total params: 769
 Trainable params: 769
 Non-trainable params: 0
 =====

Figura 31. Modelo secuencial de la red neuronal de Bimbo.

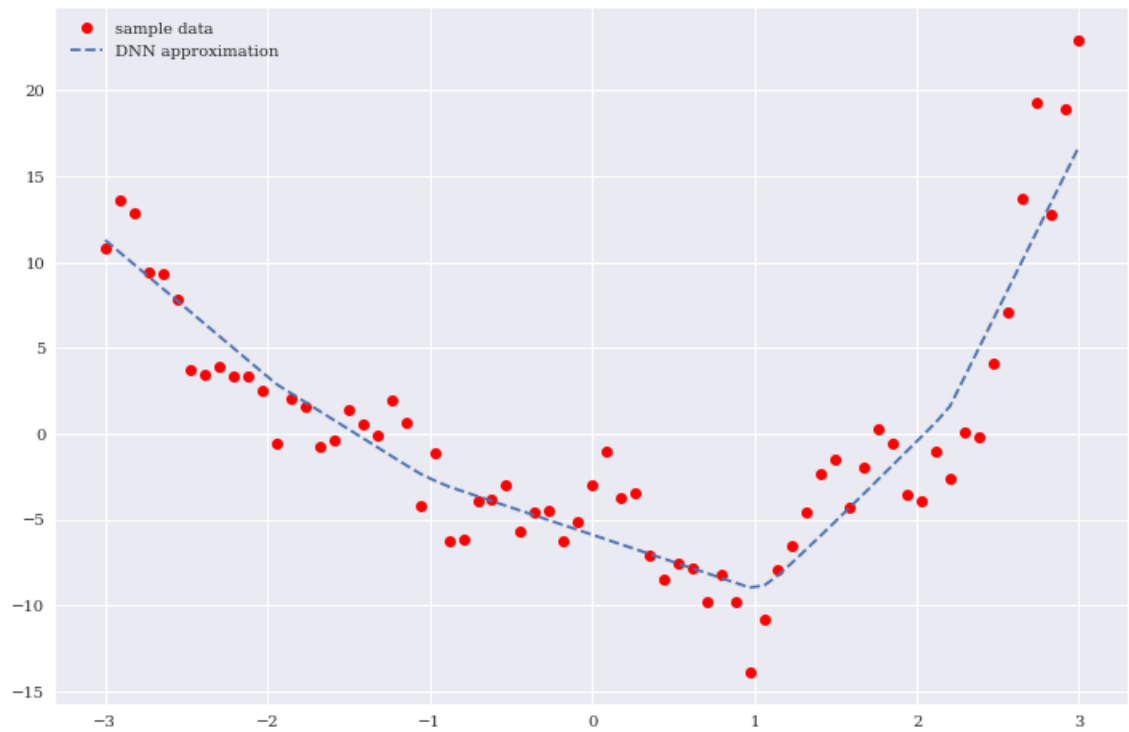


Figura 32. Aproximación de la muestra de la red neuronal profunda (DNN) de Bimbo.

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	512
dense_1 (Dense)	(None, 1)	257

Total params: 769
 Trainable params: 769
 Non-trainable params: 0

Figura 33. Modelo secuencial de la red neuronal de Amazon.

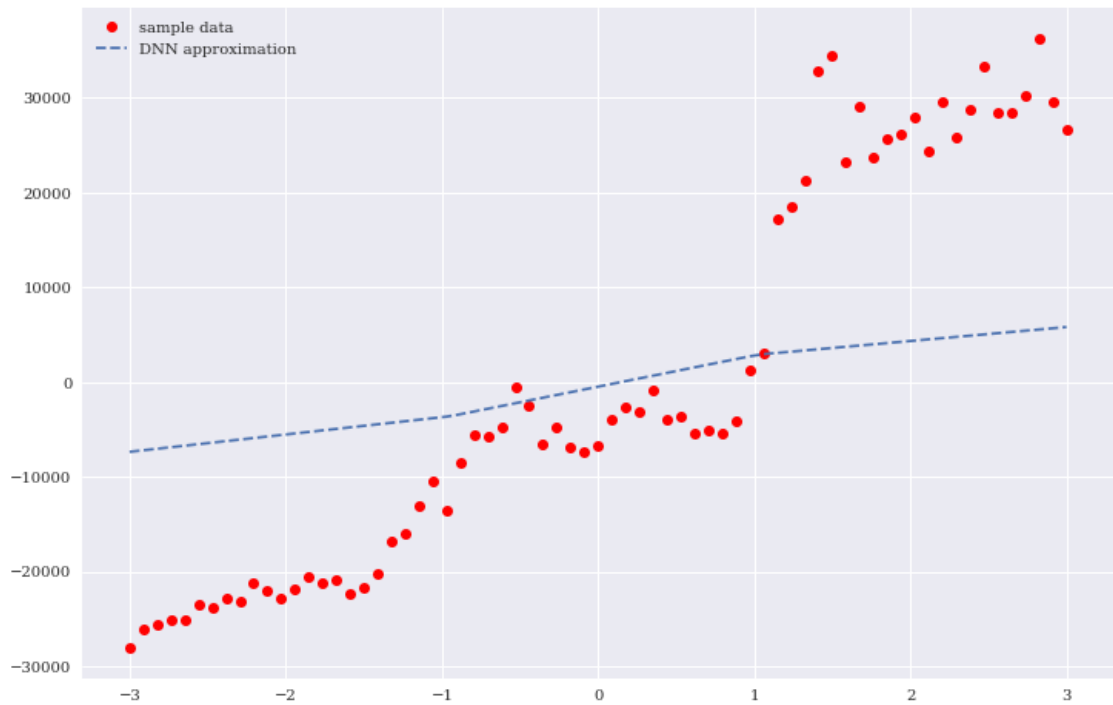


Figura 34. Aproximación de la muestra de la red neuronal profunda (DNN) de Amazon.

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	512
dense_1 (Dense)	(None, 1)	257
Total params: 769		
Trainable params: 769		
Non-trainable params: 0		

Figura 35. Modelo secuencial de la red neuronal de BanBajio.

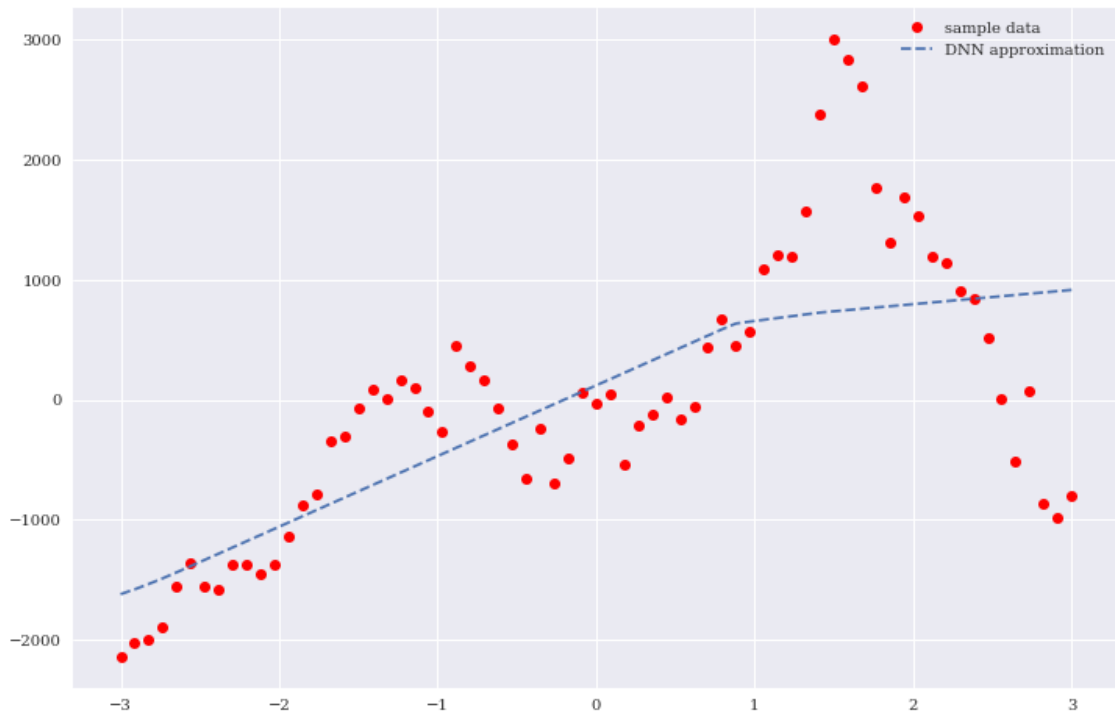


Figura 36. Aproximación de la muestra de la red neuronal profunda (DNN) de BanBajio.

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	512
dense_1 (Dense)	(None, 1)	257
Total params: 769		
Trainable params: 769		
Non-trainable params: 0		

Figura 37. Modelo secuencial de la red neuronal del Oro.

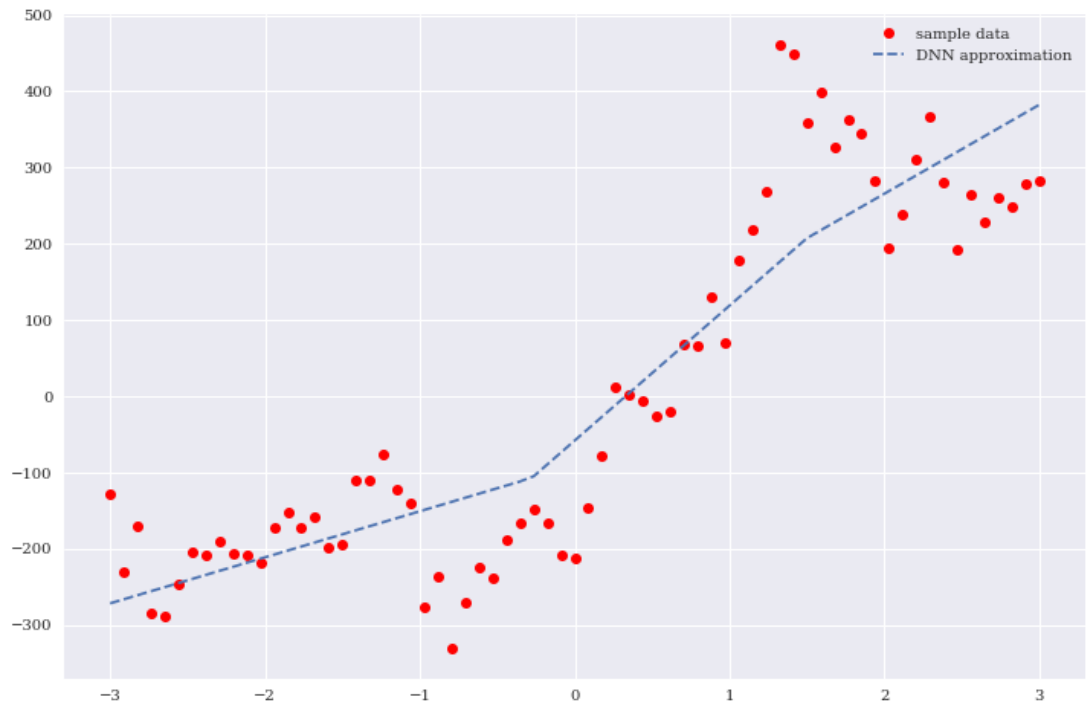


Figura 38. Aproximación de la muestra de la red neuronal profunda (DNN) del Oro.

Con el paquete Keras, los valores MSE se almacenan después de cada paso de aprendizaje. el valor de MSE ("pérdida") disminuye en promedio con el número creciente de épocas durante las cuales se entrena la red neuronal. No hay gráfica de pérdidas.

Capacidad

Cuanto mayor sea el grado del parámetro, mayor será la capacidad del modelo de regresión OLS. El código de Python comenzó en $grados = 1$ y aumenta el grado en incrementos de dos. Los valores de MSE disminuyeron monótonamente con el parámetro de grado creciente. Las Figuras 39, 40, 41, 42, y 43 muestra las líneas de regresión para todos los grados considerados de cada acción.

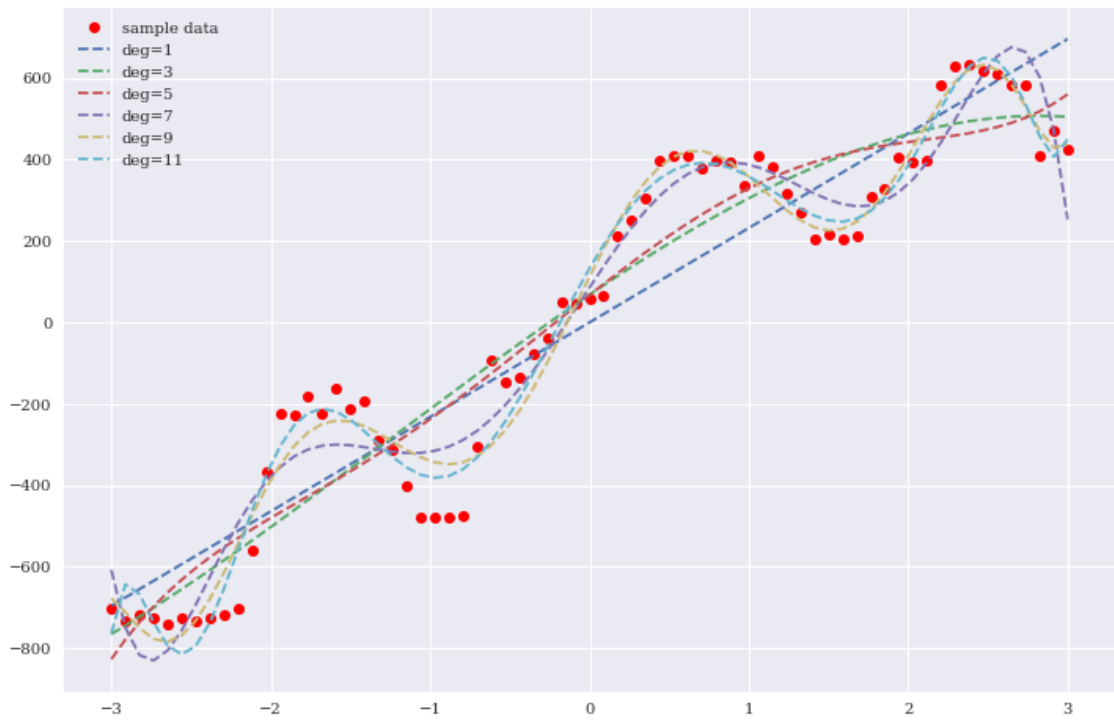


Figura 39. Muestra las líneas de regresión para todos los grados considerados de Elektra.

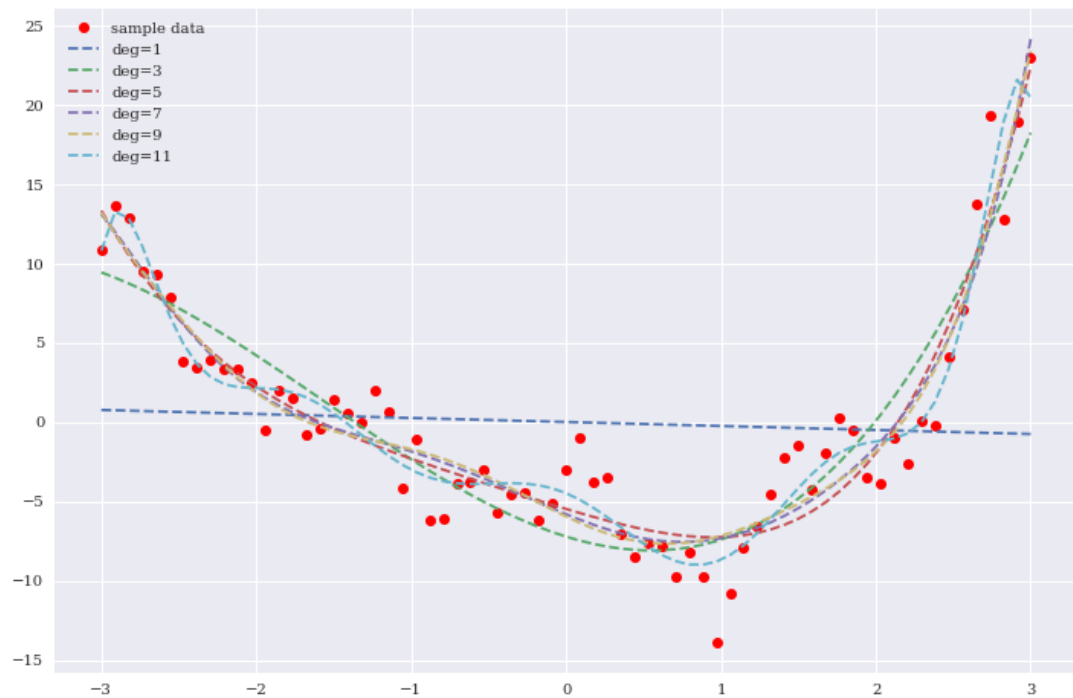


Figura 40. Muestra las líneas de regresión para todos los grados considerados de Bimbo.

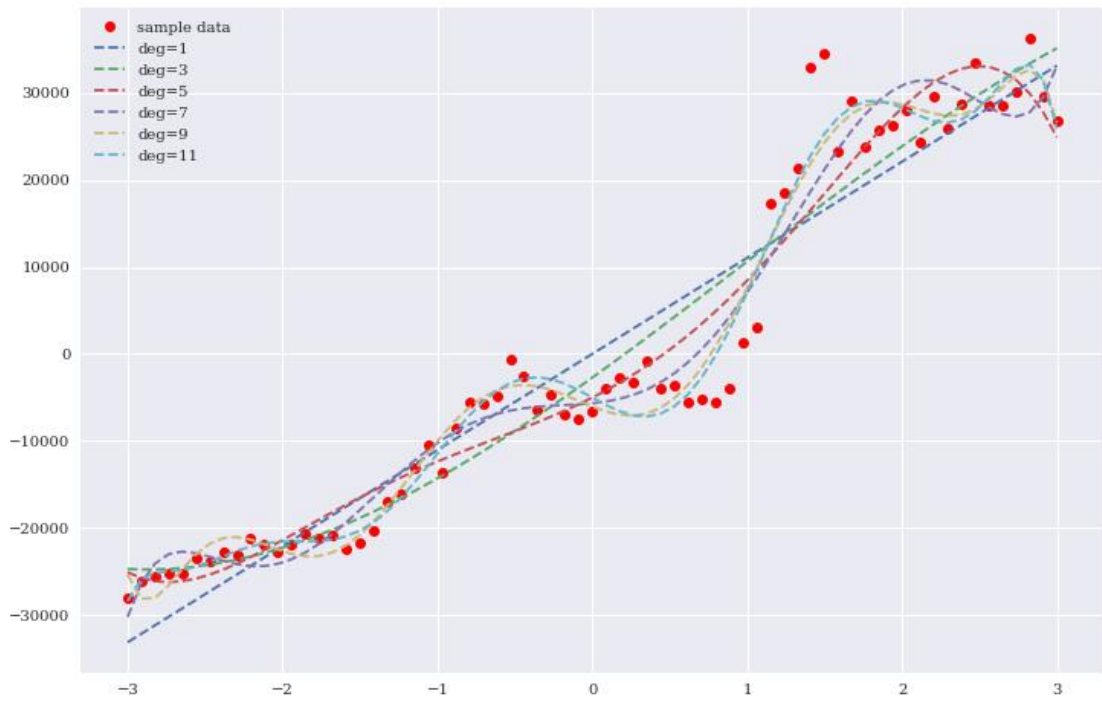


Figura 41. Muestra las líneas de regresión para todos los grados considerados de Amazon.

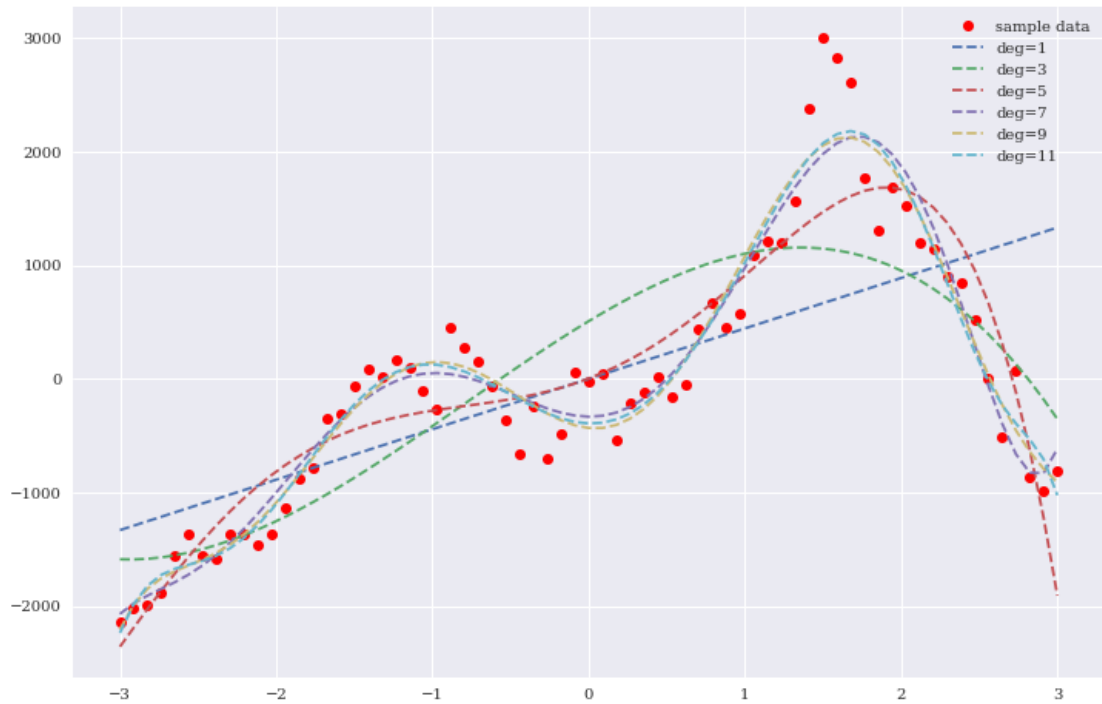


Figura 42. Muestra las líneas de regresión para todos los grados considerados de BanBajío.

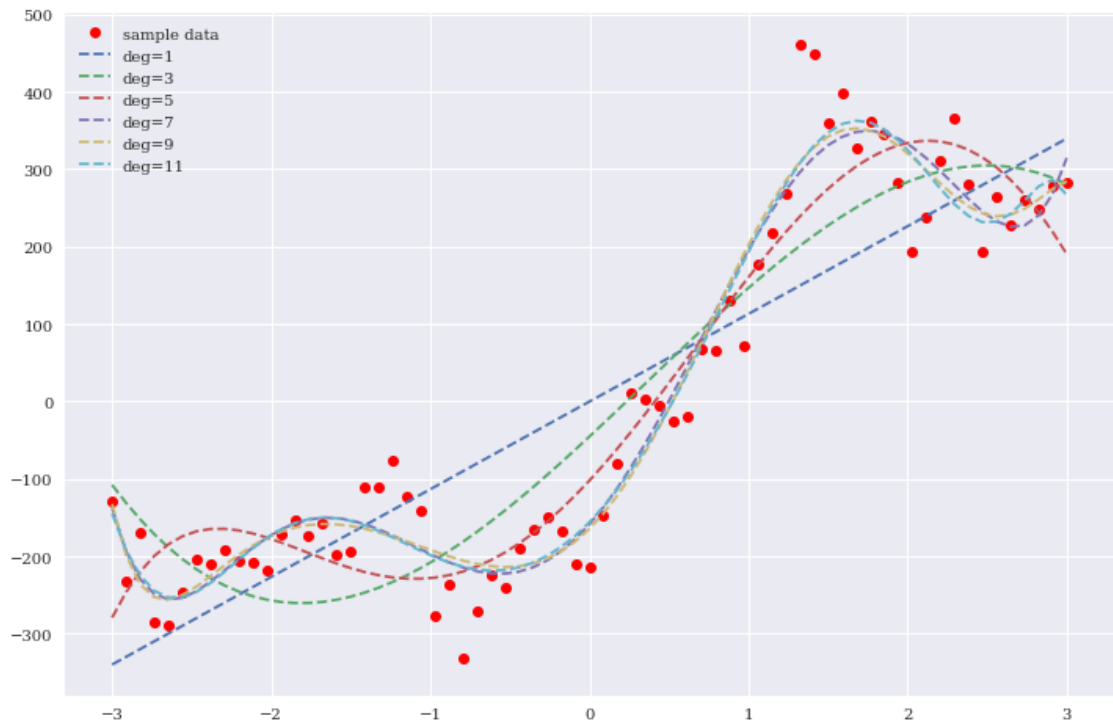


Figura 43. Muestra las líneas de regresión para todos los grados considerados del Oro.

Agregado a esto, por ejemplo; figura 44, una capa más del mismo tamaño aumenta significativamente la cantidad de parámetros entrenables.

```

Model: "sequential_1"

```

Layer (type)	Output Shape	Param #
dense_2 (Dense)	(None, 256)	512
dense_3 (Dense)	(None, 256)	65792
dense_4 (Dense)	(None, 256)	65792
dense_5 (Dense)	(None, 1)	257

```

Total params: 132,353
Trainable params: 132,353
Non-trainable params: 0

```

Figura 44. Modelo secuencial usando más capas.

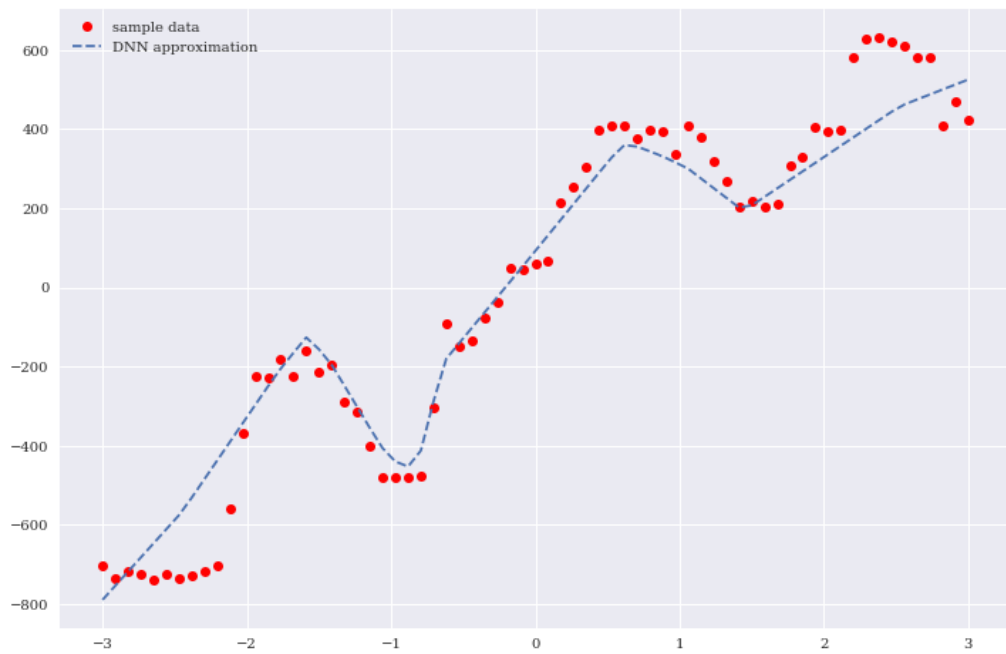


Figura 45. Aproximación de la muestra de la red neuronal profunda “con más capas” (DNN) de Elektra.

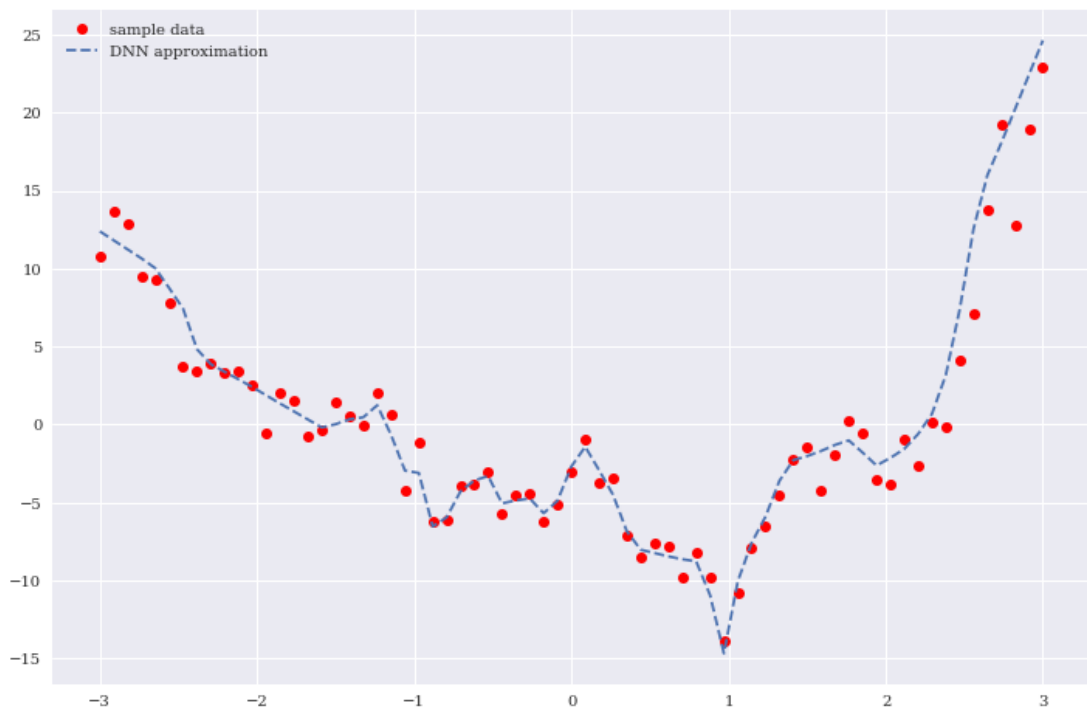


Figura 46. Aproximación de la muestra de la red neuronal profunda “con más capas” (DNN) de Bimbo.

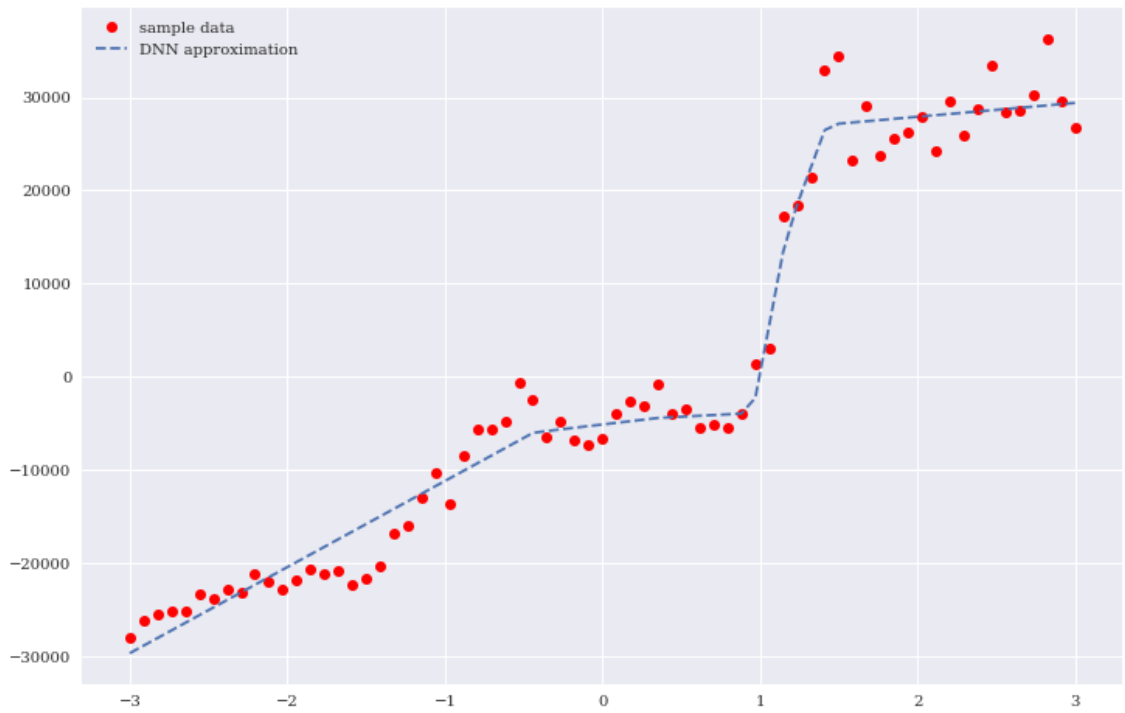


Figura 47. Aproximación de la muestra de la red neuronal profunda “con más capas” (DNN) de Amazon.

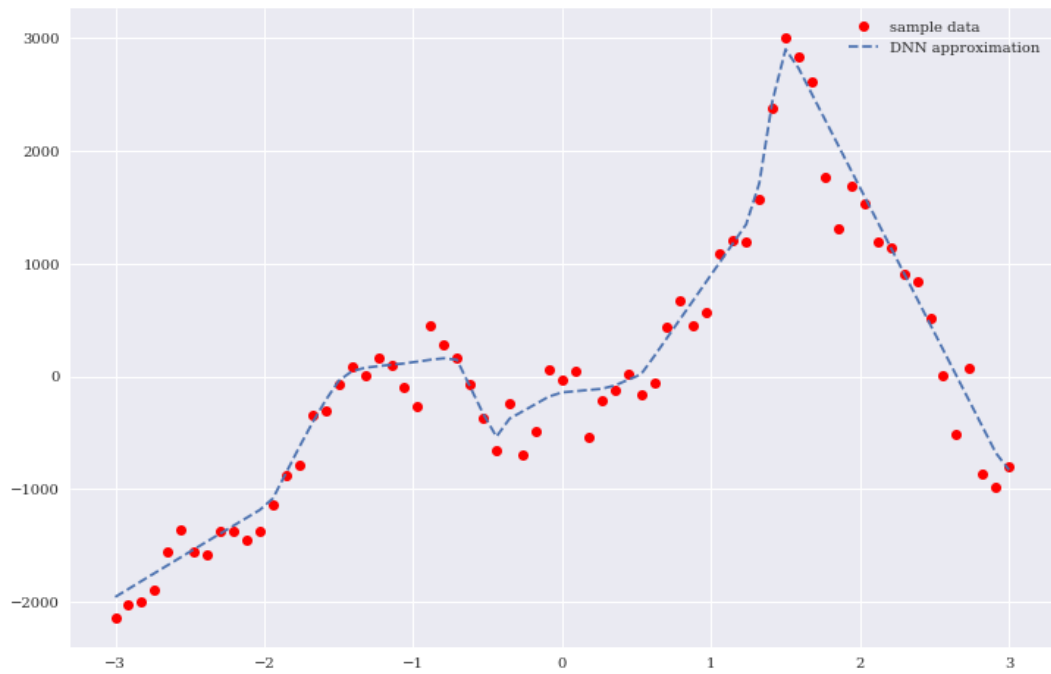


Figura 48. Aproximación de la muestra de la red neuronal profunda “con más capas” (DNN) de BanBajo.

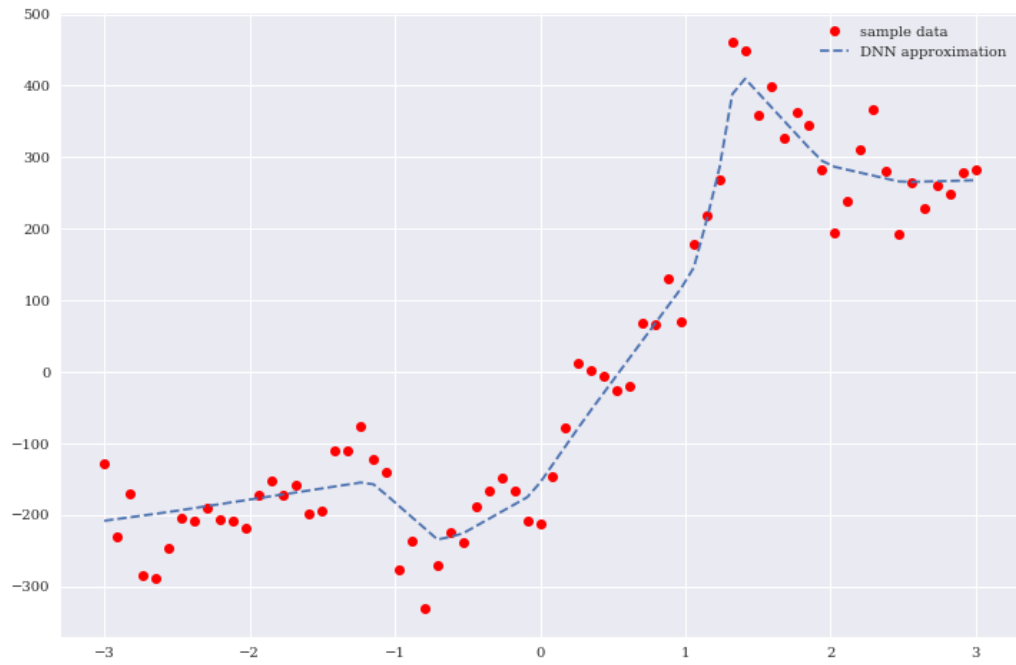


Figura 49. Aproximación de la muestra de la red neuronal profunda “con más capas” (DNN) del Oro.

Evaluación.

Las siguientes figuras 50, 51, 52, 53 y 54 muestran los ajustes de regresión para los diferentes valores de grados y comparan los ajustes para la cantidad de información de enseñanza y verificación.

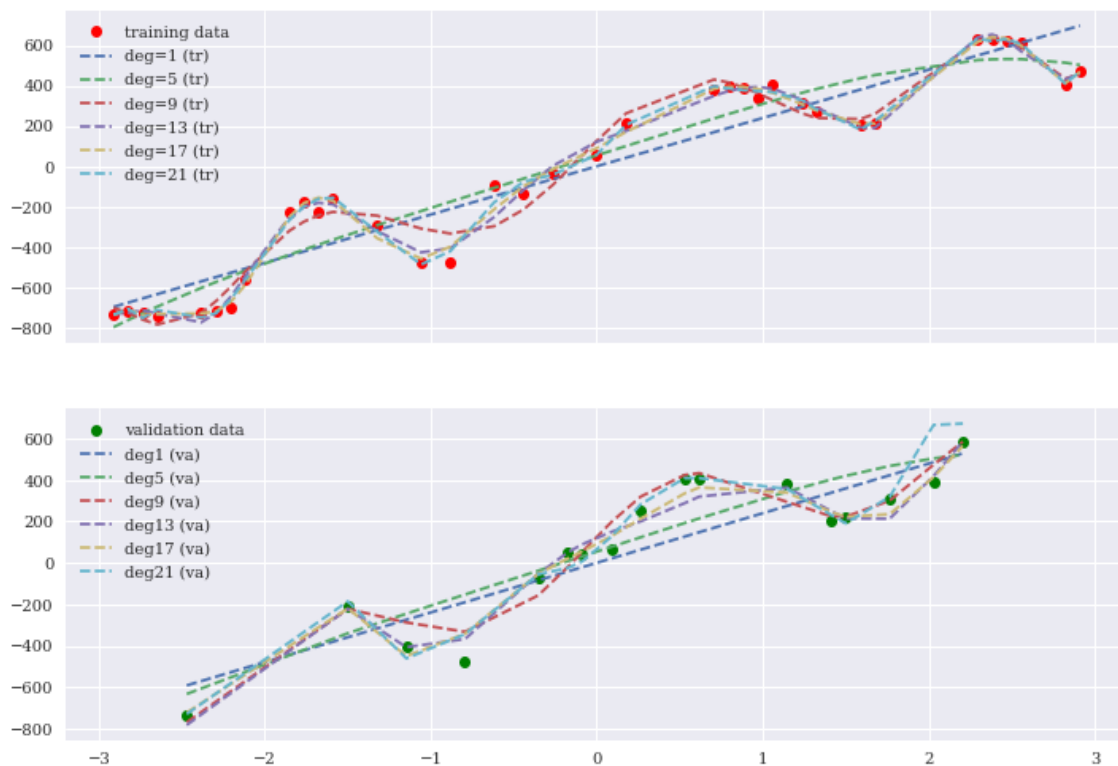


Figura 50. Ajustes de regresión de Elektra.

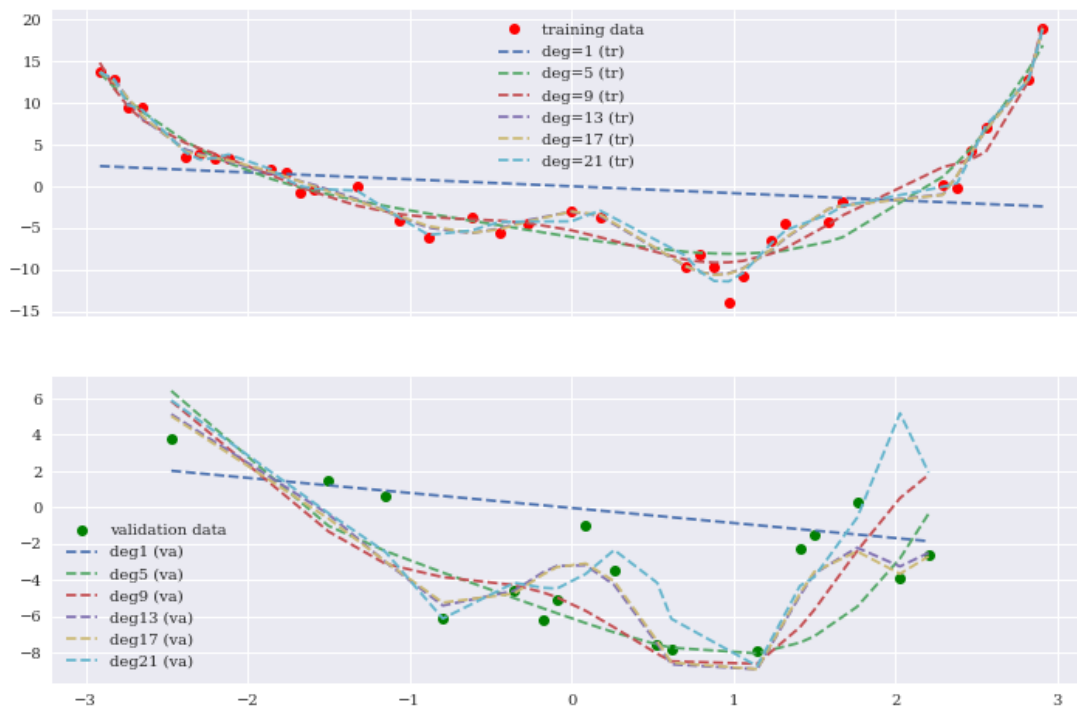


Figura 51. Ajustes de regresión de Bimbo.

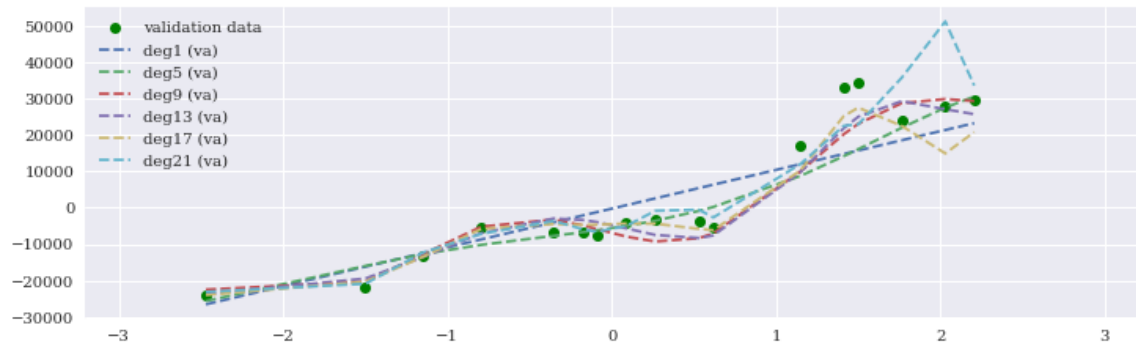
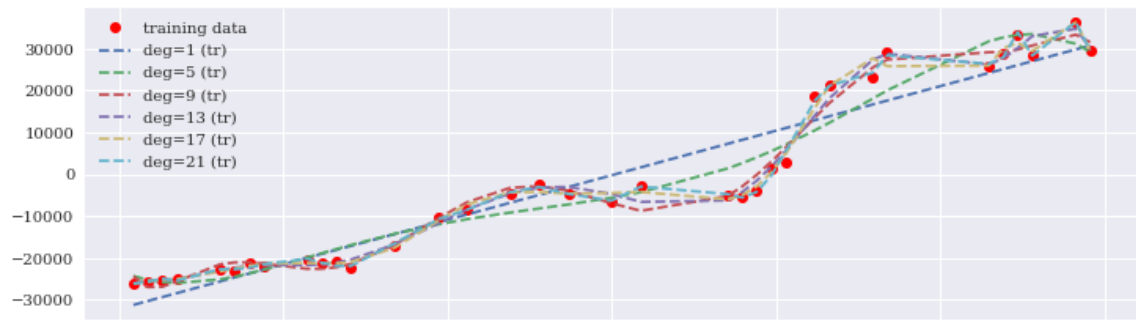


Figura 52. Ajustes de regresión de Amazon.

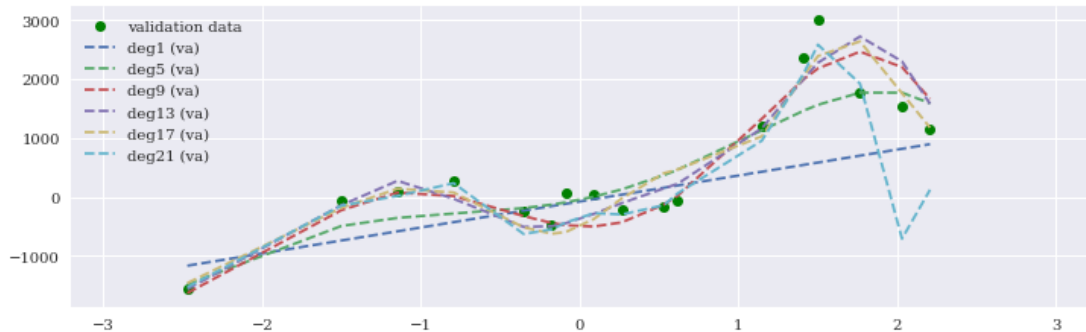
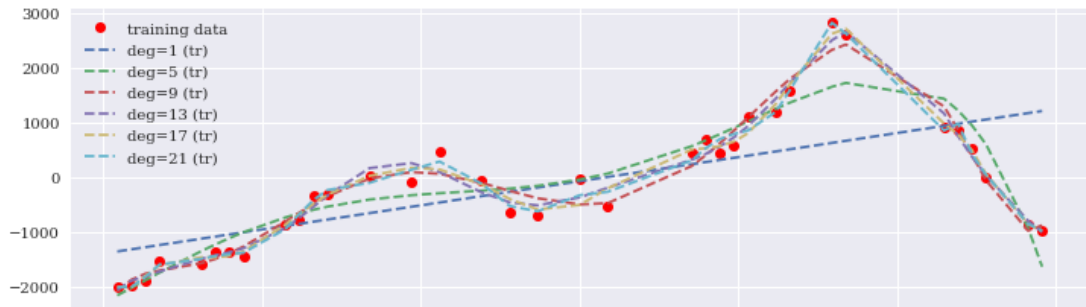


Figura 53. Ajustes de regresión de BanBajo.

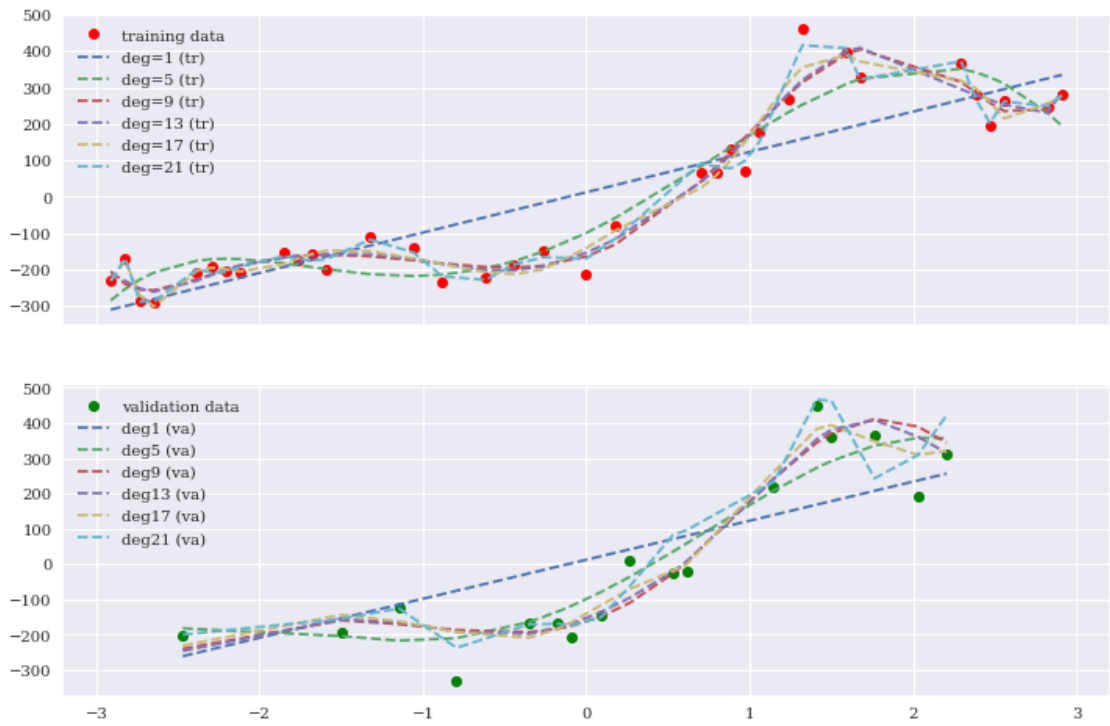


Figura 54. Ajustes de regresión del Oro.

Las siguientes Figuras 55, 56, 57, 58 y 59; muestra las predicciones de la red neuronal para la cantidad de valores de validación y entrenamiento.

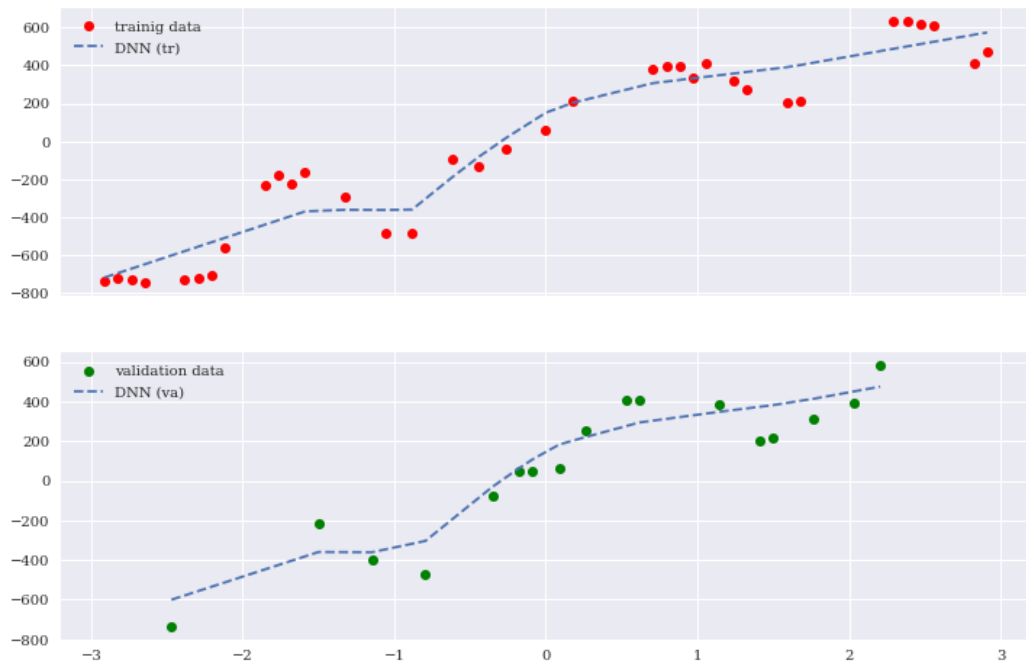


Figura 55. Muestra las predicciones de la red neuronal de la gráfica de Elektra.

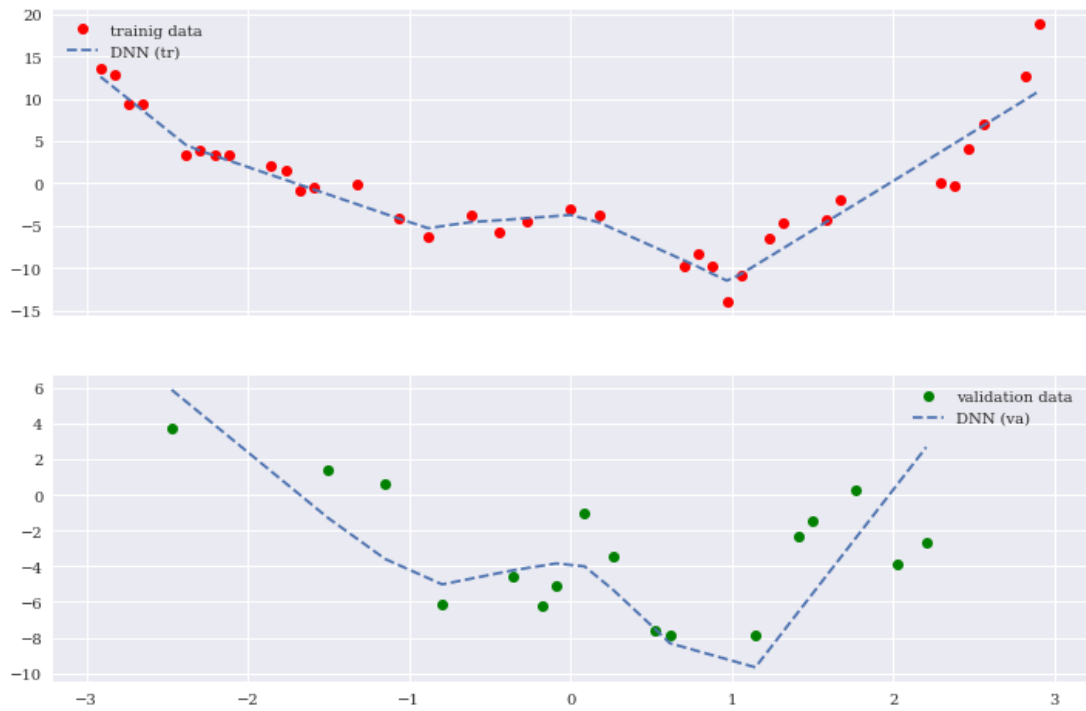


Figura 56. Muestra las predicciones de la red neuronal de la gráfica de Bimbo.

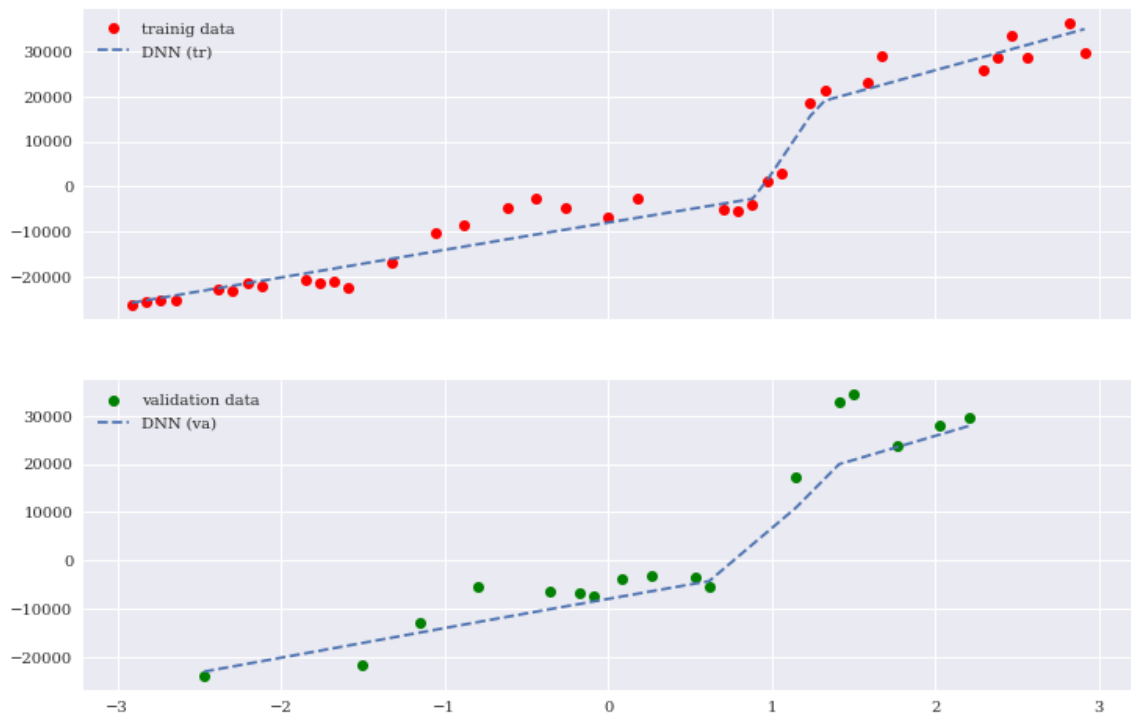


Figura 57. Muestra las predicciones de la red neuronal de la gráfica de Amazon.

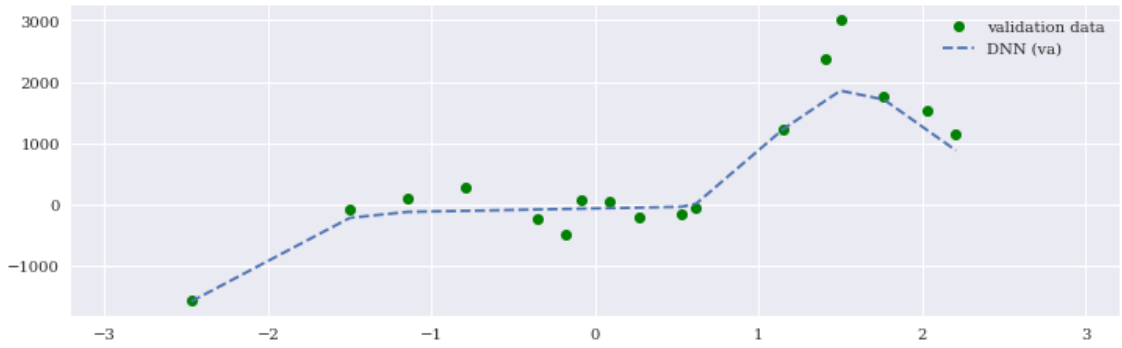
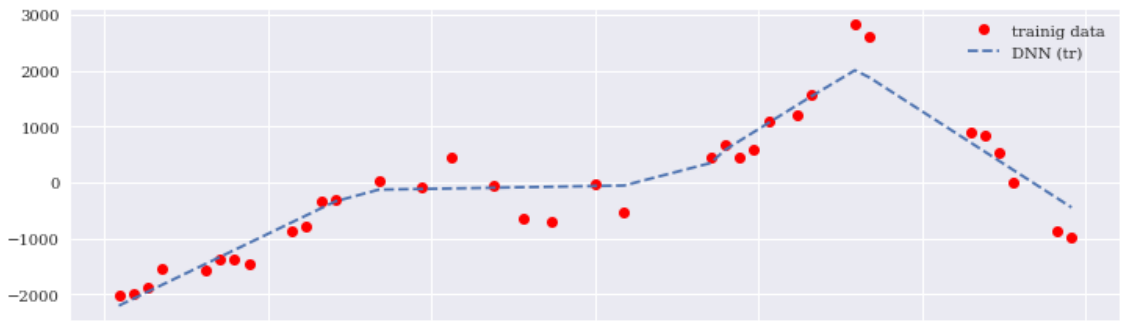


Figura 58. Muestra las predicciones de la red neuronal de la gráfica de BanBajo.

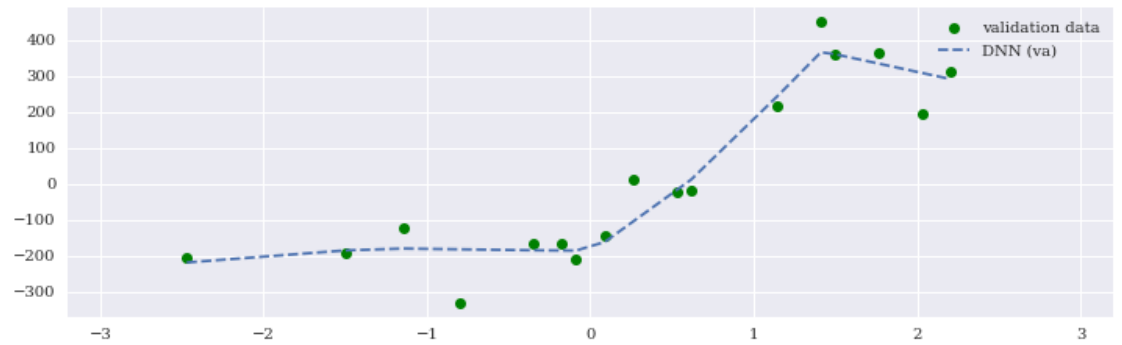
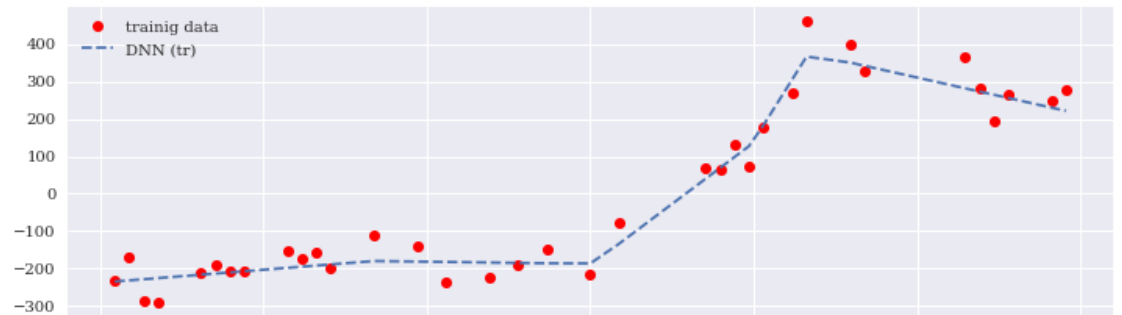


Figura 59. Muestra las predicciones de la red neuronal de la gráfica del Oro.

Las figuras 60, 61, 62, 63 y 64 compara las predicciones de ambos modelos entre sí y con el conjunto de datos de prueba.

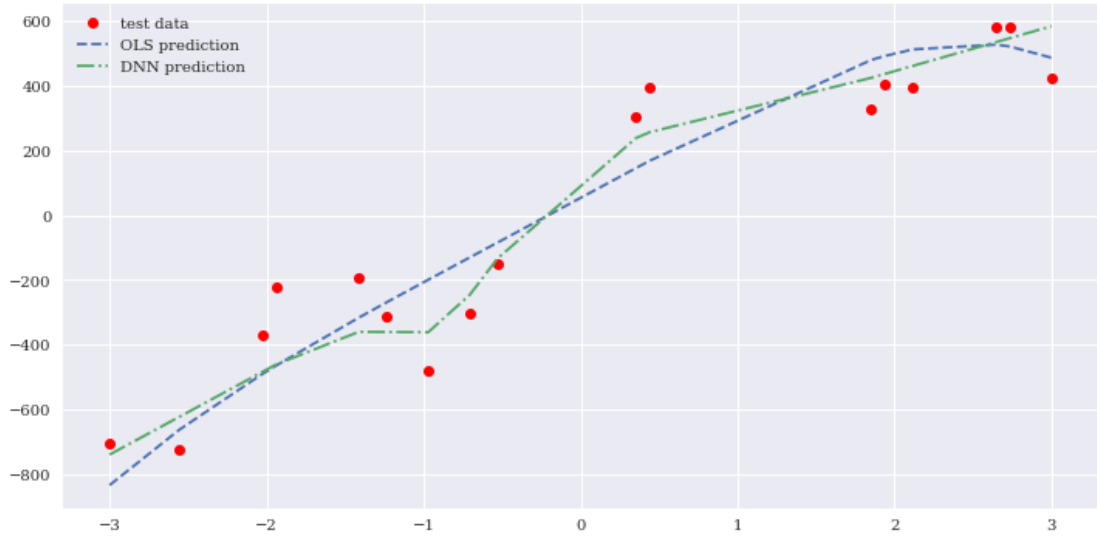


Figura 60. Medición de las predicciones de ambos modelos, Elektra.

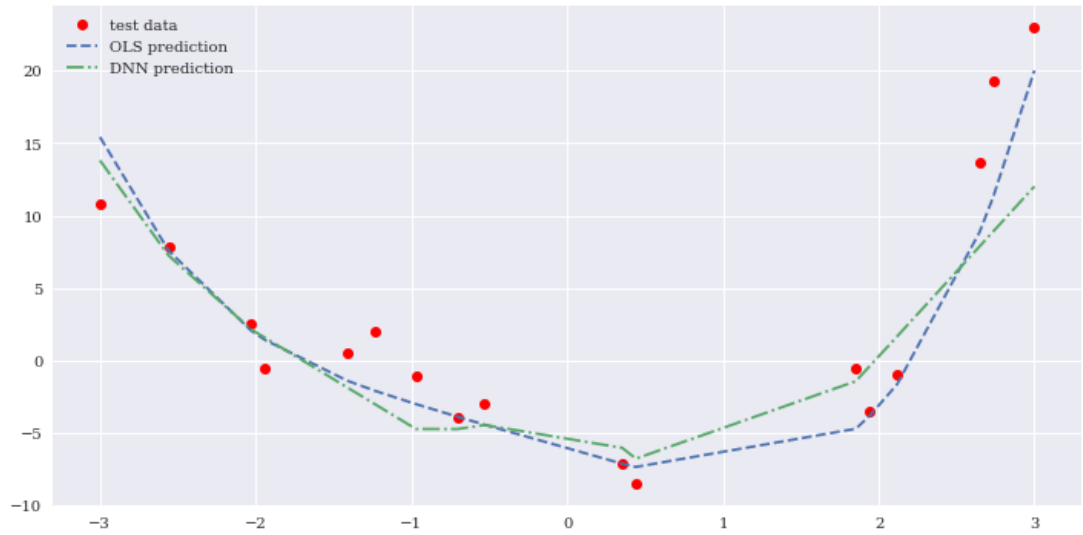


Figura 61. Medición de las predicciones de ambos modelos, Bimbo.

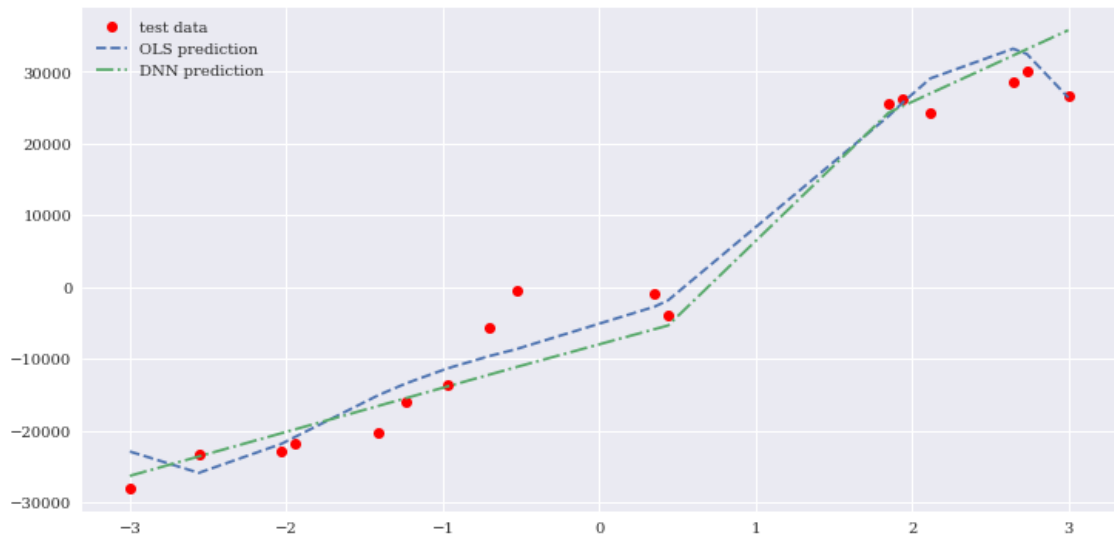


Figura 62. Comparación de las predicciones de ambos modelos, Amazon.

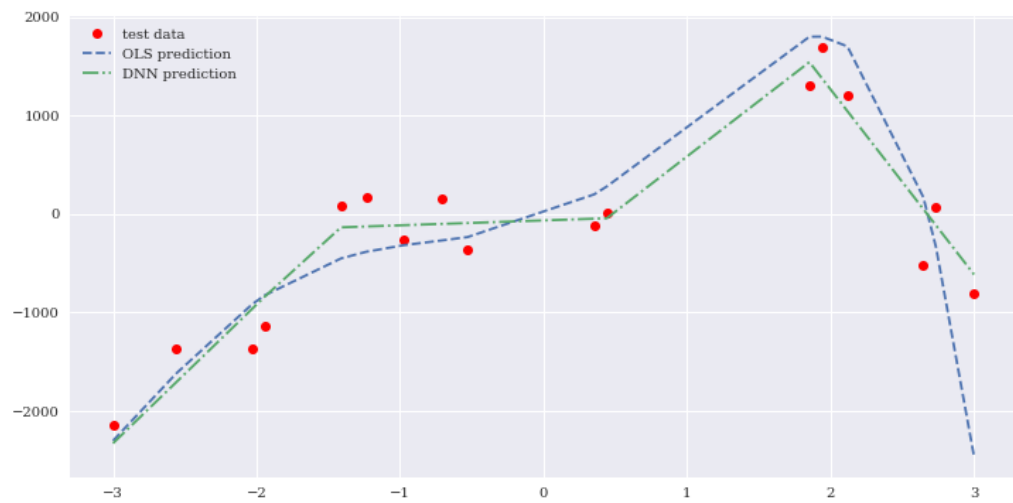


Figura 63. Comparación de las predicciones de ambos modelos, BanBajo.

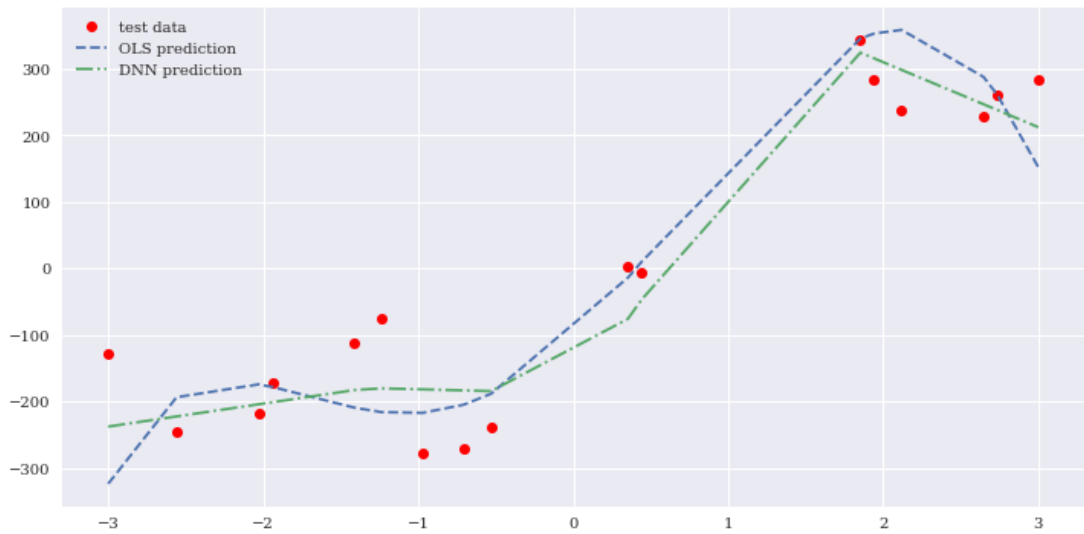


Figura 64. Comparación de las predicciones de ambos modelos, Oro.

Sesgo y varianza

El ajuste de regresión lineal, como se muestra en las figuras 65, 66, 67, 68, y 69 tiene un alto sesgo en la información de enseñanza; las variables absolutas sobre las predicciones y los datos de las etiquetas son relativamente altas. El ajuste de orden superior muestra una varianza alta. Alcanza exactamente todos los puntos de datos de entrenamiento, pero el ajuste en sí varía significativamente para lograr el ajuste perfecto. muestra que un ajuste de alto sesgo funciona peor en el ejemplo que un ajuste de alta varianza en los datos de entrenamiento.

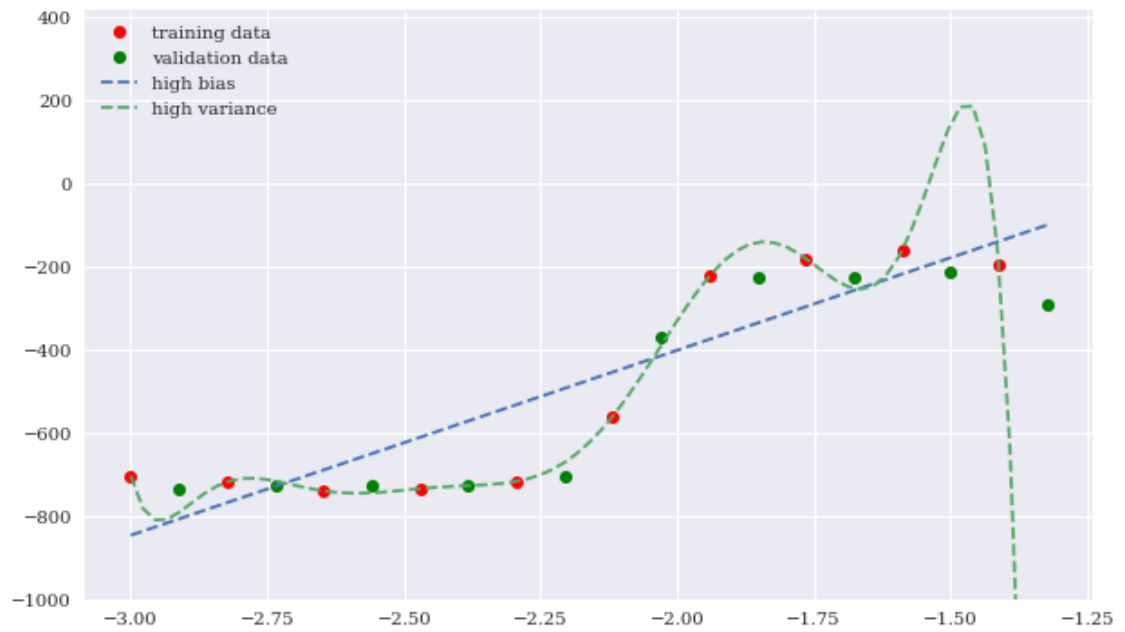


Figura 65. Ajustes de regresión OLS de alto sesgo y alta varianza, Elektra.

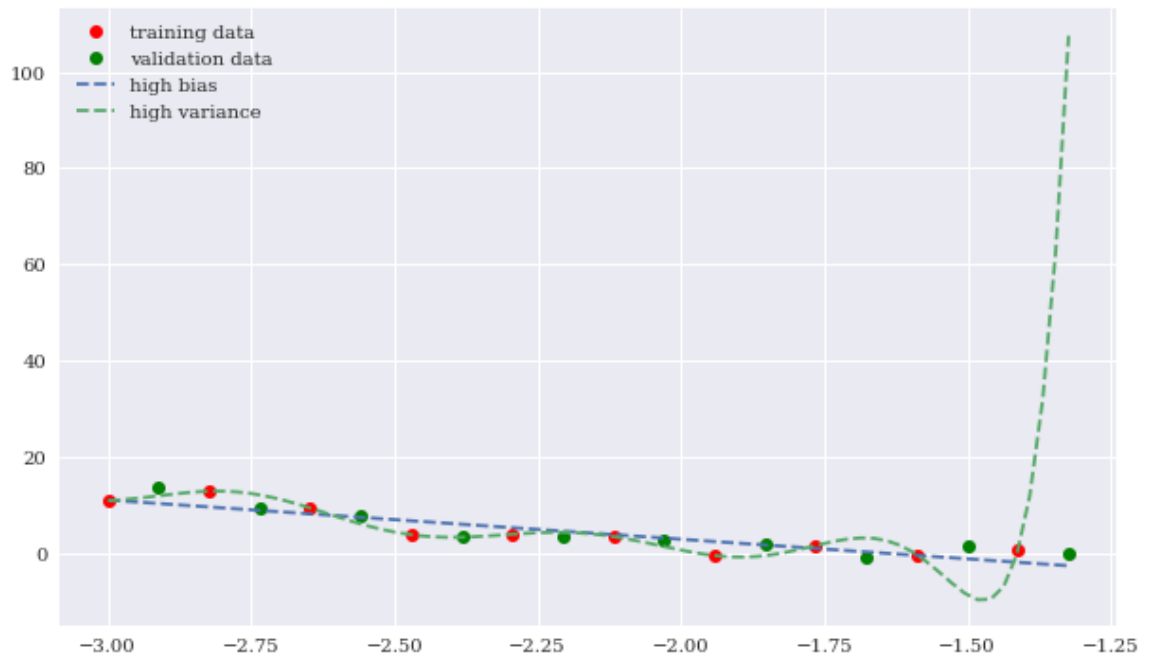


Figura 66. Ajustes de regresión OLS de alto sesgo y alta varianza, Bimbo.

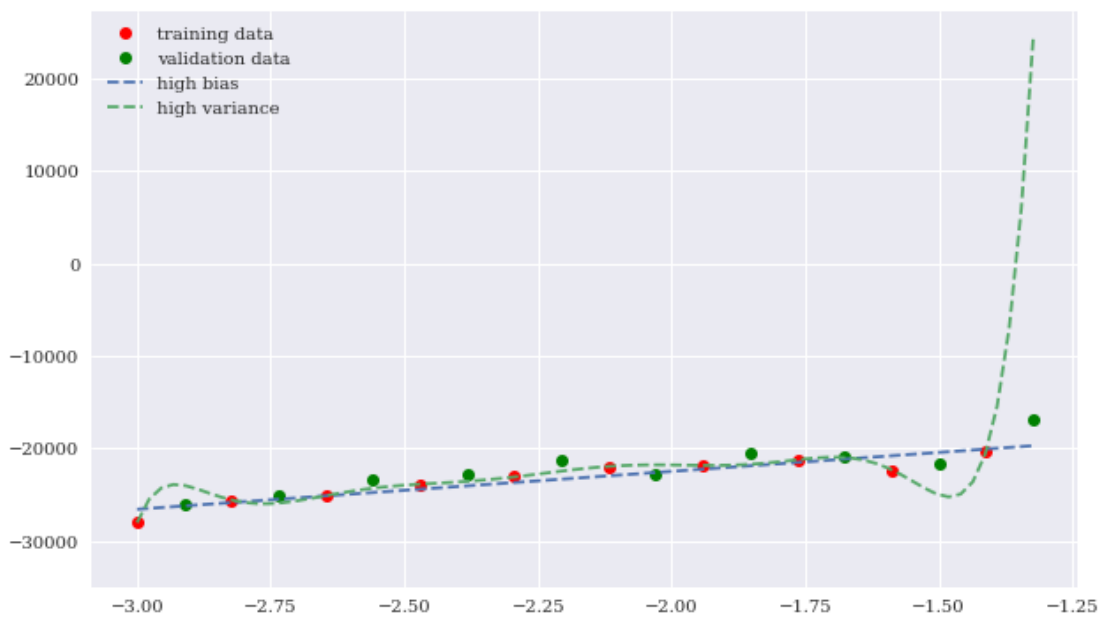


Figura 67. Ajustes de regresión OLS de alto sesgo y alta varianza, Amazon.

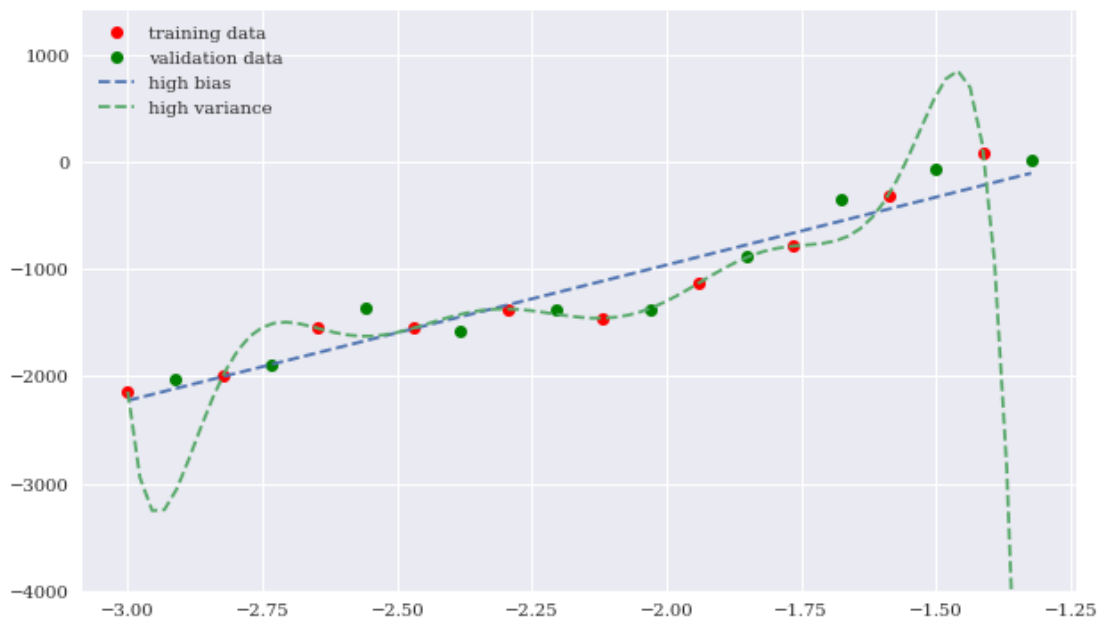


Figura 68. Ajustes de regresión OLS de alto sesgo y alta varianza, BanBajo.

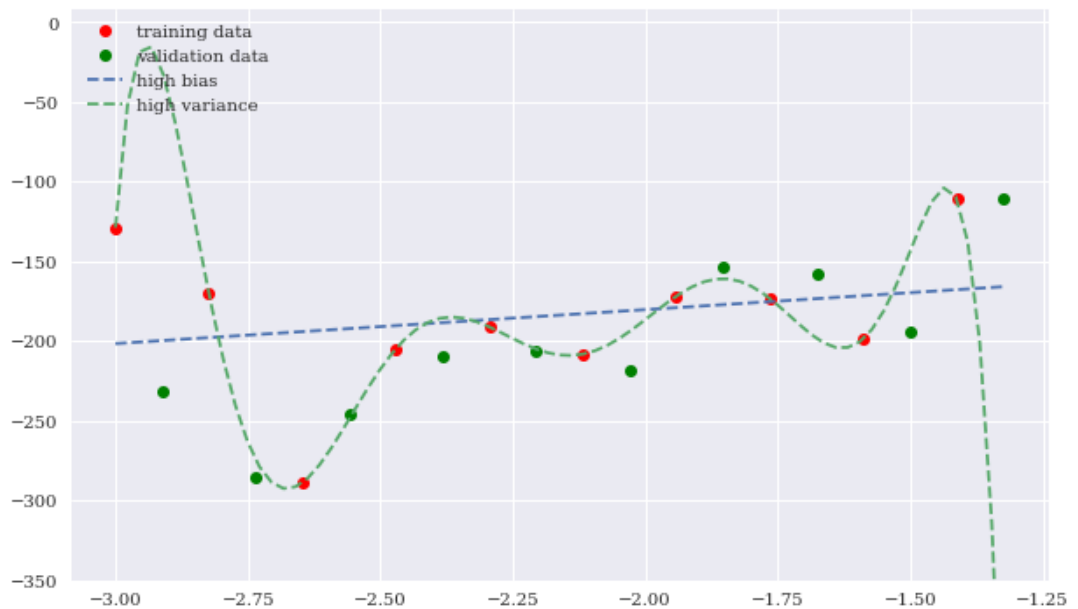


Figura 69. Ajustes de regresión OLS de alto sesgo y alta varianza, Oro.

Los resultados enseñan sobre el rendimiento del ejemplo de alto sesgo es aproximadamente comparable con la información de enseñanza y validación. Por el contrario, el rendimiento del modelo de alta varianza es perfecto en los valores de adiestramiento y bastante malo en los datos de validación.

Validación cruzada

Cross-validation scores

deg	0	1	2	3	4	5	6	7	8	9
deg=0	-17.16	-8.63	-0.71	-27.66	-43.09					
deg=1	-0.07	-1.04	0.05	-2.82	-2.49					
deg=2	0.47	-2.03	0.41	-2.11	-0.21					
deg=3	-3.72	-1.89	0.45	-4.14	-22.27					
deg=4	-116.82	-1.98	0.15	-6.27	-254.53					
deg=5	-860.06	-2.32	0.18	-8.27	-202.30					
deg=6	-315.53	-1.99	0.03	-16.77	-7802.85					
deg=7	-3472.39	-0.40	0.54	-6.79	-2864.38					
deg=8	-3.39	-0.20	0.78	-5.32	-13704.69					
deg=9	-128992.08	-4.34	0.76	-1.45	-281194.36					

Figura. 70.

Mercados eficientes

Se graficó la información desde el año 2016 hasta el 2022, véase la figura 71.



Figura.71. Información de series de tiempo normalizada.

Las figuras 72 y 73 muestran los resultados de regresión óptimos promedio. Sin duda, el dato de precios con un desfase de solo un día tiene el mayor poder explicativo. Su peso es cercano a 1, lo que respalda la idea de que el mejor predictor del valor de mañana de una herramienta financiera es su valor de hoy. Esto también es válido para los resultados de regresión simple obtenidos por serie temporal financiera:

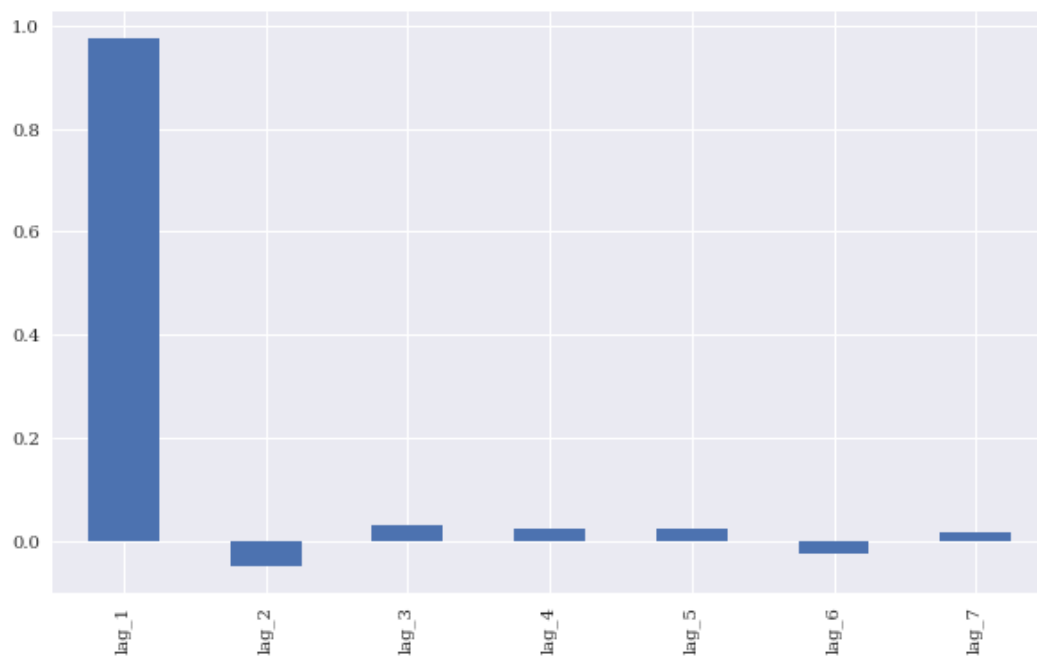


Figura.72. Promedio de parámetros de regresión óptimos para los precios rezagados.

	lag_1	lag_2	lag_3	lag_4	lag_5	lag_6	lag_7
ELEKTRA	1.1459	-0.1484	-0.0338	-0.0022	0.0642	-0.0197	-0.0056
BIMBOA	0.8653	0.0301	0.0579	0.0259	0.0288	0.0353	-0.0430
AMZN	1.0259	-0.0612	-0.0106	0.0855	-0.0217	-0.0680	0.0508
BABAN	1.0117	-0.0522	0.0155	0.0619	-0.0575	0.0090	0.0115
ORO	0.8345	-0.0130	0.1213	-0.0503	0.1057	-0.0746	0.0767

Figura 73. Resultados de regresión óptimos promedio

Predicción de mercado basada en datos de devoluciones.

Primero, se implementa la regresión OLS y se generan las predicciones resultantes de la regresión. El estudio se implementó con el conjunto de información completa. Mostró que tan bien funcionan los algoritmos en la muestra. La precisión con la que la regresión OLS predice la dirección del movimiento del día siguiente es ligeramente superior al 50 %. Véase figura 74.

```

Wall time: 0 ns
OLS) | ELEKTRA | acc=0.5276
OLS) | BIMBOA | acc=0.5305
OLS) | AMZN | acc=0.5177
OLS) | BABAN | acc=0.5064
OLS) | ORO | acc=0.5397

```

Figura 74. Predicciones resultantes de la regresión.

En segundo lugar, se vuelve a realizar el mismo análisis, pero esta vez con una red neuronal de Scikit-learn como modelo para el aprendizaje y la predicción. La precisión de la predicción en la muestra es significativamente superior al 50 % en todo momento y superior al 60 % casi 70% en algunos casos, véase figura 75.

```

Wall time: 0 ns
MLP | ELEKTRA | acc=0.5552
MLP | BIMBOA | acc=0.6813
MLP | AMZN | acc=0.6969
MLP | BABAN | acc=0.6381
MLP | ORO | acc=0.5949

```

Figura 75. Predicciones resultantes de MLP.

Tercero, el mismo análisis nuevamente, pero con una red neuronal del paquete Keras. Los resultados de precisión son similares a los de MLPRegressor, pero con una precisión promedio más alta, Véase figura 76.

```

Wall time: 0 ns
DNN | ELEKTRA | acc=0.4823
DNN | BIMBOA | acc=0.4922
DNN | AMZN | acc=0.5524
DNN | BABAN | acc=0.5028
DNN | ORO | acc=0.5297

```

Figura 76. Predicciones resultantes de DNN.

Este ejemplo simple muestra que las redes neuronales pueden superar significativamente la regresión OLS en la muestra para predecir la dirección de los movimientos de precios del día siguiente. Sin embargo, se repitieron los análisis, pero el paso de entrenamiento (ajuste) se implementó en el primer 80 % de los datos mientras que el rendimiento se prueba en el 20 % restante. La regresión OLS se implementó primero; la regresión OLS fuera de la muestra

se reflejaron niveles de precisión similares a los de la muestra, alrededor del 50 %, véase figura 77.

```
Wall time: 0 ns
OLS | ELEKTRA | acc=0.5194
OLS | BIMBOA  | acc=0.4947
OLS | AMZN    | acc=0.5053
OLS | BABAN   | acc=0.5406
OLS | ORO     | acc=0.5336
```

Figura 77. Predicciones resultantes de OLS.

El rendimiento del modelo MLPRegressor fuera de la muestra es peor en comparación con los números dentro de la muestra y es similar a los resultados de la regresión OLS, véase figura 78.

```
Wall time: 0 ns
MLP | ELEKTRA | acc=0.5194
MLP | BIMBOA  | acc=0.5124
MLP | AMZN    | acc=0.5512
MLP | BABAN   | acc=0.4841
MLP | ORO     | acc=0.4735
```

Figura 78. Predicciones resultantes de MLP.

Lo mismo ocurrió con el modelo secuencial de Keras, para el cual los números fuera de la muestra también mostraron valores de precisión entre unos pocos puntos porcentuales por encima y por debajo del umbral del 50 %. Véase figura 79.

```
Wall time: 0 ns
DNN | ELEKTRA | acc=0.4735
DNN | BIMBOA  | acc=0.2933
DNN | AMZN    | acc=0.5336
DNN | BABAN   | acc=0.4205
DNN | ORO     | acc=0.4770
```

Figura 79. Predicciones resultantes de DNN.

Predicción de mercado con más características.

En la muestra, el rendimiento del modelo MLPClassifier ahora es mucho mejor cuando se tienen en cuenta las nuevas funciones y se normalizaron para el entrenamiento. El modelo secuencial de Keras alcanzó precisiones en torno al 70% para el número de épocas

entrenadas. Estos pueden incrementarse fácilmente aumentando el número de épocas y/o la capacidad de la red neuronal. Véase figura 80.

```
Wall time: 0 ns
IN-SAMPLE | ELEKTRA | acc=0.6147
IN-SAMPLE | BIMBOA | acc=0.5670
IN-SAMPLE | AMZN | acc=0.5707
IN-SAMPLE | BABAN | acc=0.6740
IN-SAMPLE | ORO | acc=0.6176
```

```
Wall time: 0 ns
IN-SAMPLE | ELEKTRA | acc=0.6864
IN-SAMPLE | BIMBOA | acc=0.7568
IN-SAMPLE | AMZN | acc=0.7011
IN-SAMPLE | BABAN | acc=0.7458
IN-SAMPLE | ORO | acc=0.7480
```

Figura 80.

Se repitió el análisis, esta vez con la división de entrenamiento y prueba como se usó antes. Desafortunadamente, la imagen es mixta en el mejor de los casos. Los números no representaron mejoras reales en comparación con el enfoque, ya que solo se basaron en datos de retorno retrasados como características. Para instrumentos seleccionados, pareció haber una ventaja de algunos puntos porcentuales en la precisión de la predicción en comparación con el punto de referencia del 50 %. Para otros, sin embargo; la precisión estuvo por debajo del 50%, como se ilustra para el modelo MLPClassifier. Véase la figura 81.

```
OUT-OF-SAMPLE | ELEKTRA | acc=0.5317
OUT-OF-SAMPLE | BIMBOA | acc=0.4732
OUT-OF-SAMPLE | AMZN | acc=0.4585
OUT-OF-SAMPLE | BABAN | acc=0.4098
OUT-OF-SAMPLE | ORO | acc=0.5512
Wall time: 3.35 s
```

Figura 81. Modelo MLPClassifier con la división de entrenamiento y prueba

El buen desempeño dentro de la muestra y el no tan bueno fuera de la misma; reflejó que el sobreajuste de la red neuronal desempeñó un papel crucial. Un enfoque para evitar el sobreajuste fue utilizar métodos de conjunto que combinen múltiples modelos entrenados del mismo tipo para generar un metamodelo más sólido y mejores predicciones fuera de la muestra. Uno de estos métodos se llamó embolsado. Scikit-learn tuvo una implementación

de este enfoque en forma de la clase *BaggingClassifier*. El uso de múltiples estimadores permitió entrenar a cada uno de ellos sin exponerlos al monto de información de enseñanza completo o a todas las características, esto ayudó a evitar el sobreajuste. Las precisiones de predicción estuvieron constantemente por encima del 50%. Algunos valores de precisión estuvieron por encima del 55 %, lo que pudo considerarse bastante alto en este contexto. En general, el embolsado pareció evitar, al menos hasta cierto punto, el sobreajuste y mejorar notablemente las predicciones. Véase figura 82.

```

OUT-OF-SAMPLE | ELEKTRA | acc=0.5317
OUT-OF-SAMPLE | BIMBOA  | acc=0.4537
OUT-OF-SAMPLE | AMZN   | acc=0.5366
OUT-OF-SAMPLE | BABAN  | acc=0.5561
OUT-OF-SAMPLE | ORO    | acc=0.5463
Wall time: 16.6 s

```

Figura 82. Valores fuera de la muestra.

Predicción de mercado intradía.

La implementación técnica de los datos por hora es esencialmente la misma que antes, y se basó en el mismo código que el análisis al final del día, véase figura 83.

```

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1420 entries, 2016-05-27 to 2022-01-14
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   ELEKTRA     1420 non-null   float64
1   BIMBOA      1420 non-null   float64
2   AMZN        1420 non-null   float64
3   BABAN       1420 non-null   float64
4   ORO         1420 non-null   float64
dtypes: float64(5)
memory usage: 66.6 KB
OUT-OF-SAMPLE | ELEKTRA | acc=0.5317
OUT-OF-SAMPLE | BIMBOA  | acc=0.4732
OUT-OF-SAMPLE | AMZN    | acc=0.4585
OUT-OF-SAMPLE | BABAN   | acc=0.4098
OUT-OF-SAMPLE | ORO     | acc=0.5512
Wall time: 3.02 s

```

Figura 83. Valores de la predicción de mercado intradía.

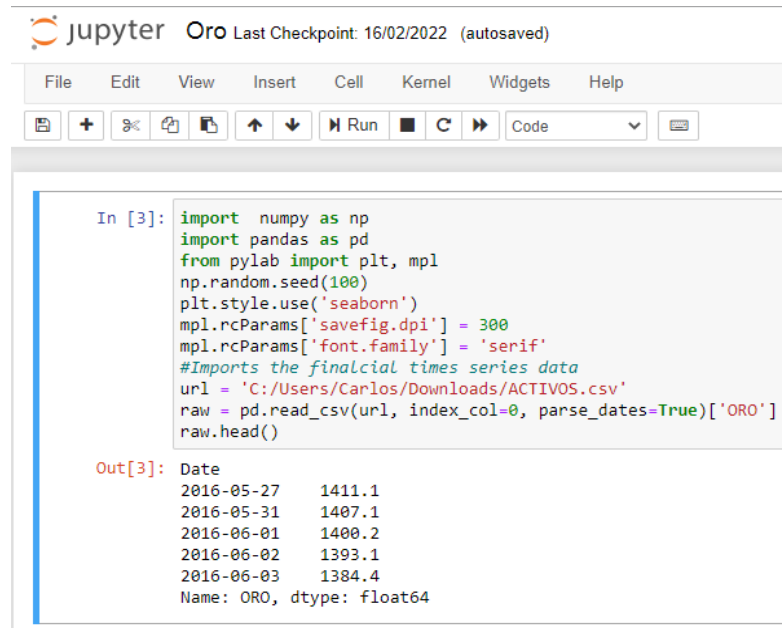
Las precisiones de predicción intradía se distribuyeron nuevamente alrededor del 50% con una dispersión relativamente amplia para la red neuronal única. En el lado positivo, algunos valores de precisión estuvieron por encima del 55%. Sin embargo, el metamodelo de embolsado mostró un rendimiento fuera de la muestra más consistente, con muchos de los valores de precisión observados unos pocos puntos porcentuales por encima del punto de referencia del 50 %. Véase figura 84.

```
OUT-OF-SAMPLE | ELEKTRA | acc=0.5317
OUT-OF-SAMPLE | BIMBOA | acc=0.4537
OUT-OF-SAMPLE | AMZN | acc=0.5366
OUT-OF-SAMPLE | BABAN | acc=0.5561
OUT-OF-SAMPLE | ORO | acc=0.5463
Wall time: 10.9 s
```

Figura 84.

Código

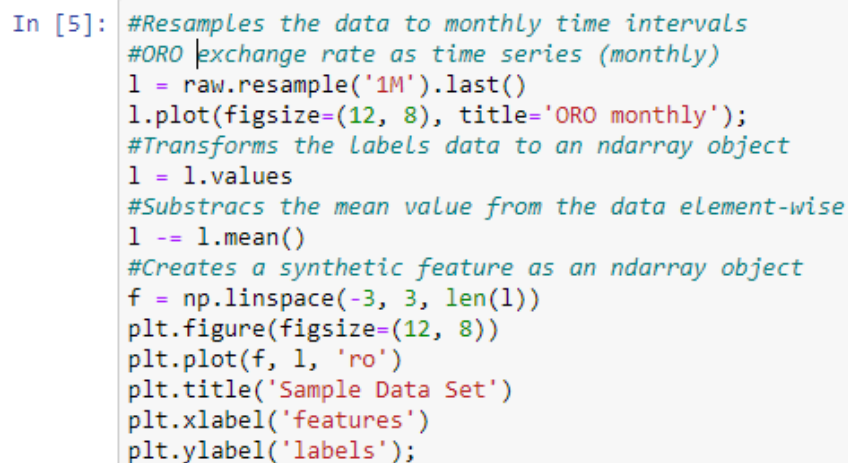
A continuación, se mostrarán algunas partes del código en forma de imágenes, con el fin de que el lector pueda darse idea del nivel de complejidad y la manera en cómo está escrito [10].



```
Jupyter Oro Last Checkpoint: 16/02/2022 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
+ < > Run Code
In [3]: import numpy as np
import pandas as pd
from pylab import plt, mpl
np.random.seed(100)
plt.style.use('seaborn')
mpl.rcParams['savefig.dpi'] = 300
mpl.rcParams['font.family'] = 'serif'
#Imports the financial times series data
url = 'C:/Users/Carlos/Downloads/ACTIVOS.csv'
raw = pd.read_csv(url, index_col=0, parse_dates=True)['ORO']
raw.head()

Out[3]: Date
2016-05-27    1411.1
2016-05-31    1407.1
2016-06-01    1400.2
2016-06-02    1393.1
2016-06-03    1384.4
Name: ORO, dtype: float64
```

Figura 85. Inicio de código, declaración de librerías, y la extracción de la información de los precios del Oro [10].



```
In [5]: #Resamples the data to monthly time intervals
#ORO exchange rate as time series (monthly)
l = raw.resample('1M').last()
l.plot(figsize=(12, 8), title='ORO monthly');
#Transforms the labels data to an ndarray object
l = l.values
#Substracs the mean value from the data element-wise
l -= l.mean()
#Creates a synthetic feature as an ndarray object
f = np.linspace(-3, 3, len(l))
plt.figure(figsize=(12, 8))
plt.plot(f, l, 'ro')
plt.title('Sample Data Set')
plt.xlabel('features')
plt.ylabel('labels');
```

Figura 86. Se muestran las descripciones del funcionamiento de las líneas para poder graficar [10].

```

In [6]: #Success
#The funtion MSE calculates the mean-squared error.
def MSE(l, p):
    return np.mean((l - p) ** 2)
#The fitting of the OLS regression model up to and including fifth-order monomials.
reg = np.polyfit(f, l, deg=5)
reg

Out[6]: array([ 1.52445863, -7.7317958 , -29.75346094,  75.98737231,
                222.51279926, -101.97267404])

In [7]: #out
#array([-0.01910626, -0.0147182 ,  0.10990388,  0.06007211, -0.20833598, -0.03275423])
#The prediction by the OLS regression model given the optimal parameters.
p = np.polyval(reg, f)
#The MSE value given the prediction values
MSE(l, p)

Out[7]: 5432.780560350674

In [8]: #Sample data and cubic regression line
plt.figure(figsize=(12, 8))
plt.plot(f, l, 'ro', label='sample data')
plt.plot(f, p, '--', label='regression')
plt.legend();

```

Figura 87. Se muestra la descripción y el código para calcular el MSE y poder graficar puntos sintéticos [10].

```

In [9]: #Regression step based on data sub-set
for i in range(10, len(f) + 1, 20):
    reg = np.polyfit(f[:i], l[:i], deg=3)
#Prediction step bases on the complete data set
p = np.polyval(reg, f)
#Resulting MSE value
mse = MSE(l, p)
print(f'{i:3d} | MSE={mse}')

10 | MSE=8732870705.140505
30 | MSE=5734408.432455546
50 | MSE=331125.37389894197

In [10]: import tensorflow as tf
tf.random.set_seed(100)
from keras.layers import Dense
from keras.models import Sequential
#Using TensorFlow backend.
#The neural network is a shallow network with a single hideden layer
model = Sequential()
model.add(Dense(256, activation='relu', input_dim=1))
model.add(Dense(1, activation='linear'))
model.compile(loss='mse', optimizer='rmsprop')
model.summary()

```

Figura 88. Cálculo del MSE, red neuronal con una sola capa [10].


```
In [24]: #Training and validation data including DNN predictions
fig, ax = plt.subplots(2, 1, sharex=True, figsize=(12, 8))
ax[0].plot(f_tr, l_tr, 'ro', label='trainig data')
p = model.predict(f_tr)
ax[0].plot(f_tr, p, '--', label=f'DNN (tr)')
ax[0].legend()
ax[1].plot(f_va, l_va, 'go', label='validation data')
p = model.predict(f_va)
ax[1].plot(f_va, p, '--', label=f'DNN (va)')
ax[1].legend();
res = pd.DataFrame(model.history.history)
res.tail()
```

Figura 89. Validación y entrenamiento de la red neuronal [10].

Conclusiones.

Se demostró que las redes neuronales artificial pueden tener una precisión de predicción del 75% en el caso de bimbo, ya que; a pesar de la complejidad, la velocidad y la forma de procesamiento de información es muy buena, mostrando datos precisos en poco tiempo. Con esto, se amplía la cantidad de herramientas para poder hacer un análisis de portafolio de inversión y tener más datos disponibles para seleccionar la opción que mejor convenga a la hora de invertir, la intención de esta tesis no es promover ni fanatizar la mejor teoría económica, sino; hacer de toda la información disponible el mejor método para invertir, estando consciente que el mundo de las finanzas es muy grande y voluble, que constantemente salen nuevos métodos de estudio para invertir. En segundo plano es importante promover el estudio en finanzas, para mejorar la infraestructura de información y se tenga más consciencia de cómo funciona la economía.

Hoy en día no existen ecuaciones matemáticas que describan de manera elegante los comportamientos de la economía como se hace en la física, hay métodos matemáticos que proporcionan cierta información para que el inversionista tenga un panorama lo más cercano de como invertir, pero; no al cien por ciento ya que hay variables que no se pueden controlar, como es el comportamiento humano y los movimientos políticos que influyen en gran medida en las finanzas, saber qué hará la gente dentro de un año o que harán los políticos es algo que no se incluye en los cálculos matemáticos, por esta razón es complicado predecir el comportamiento de las acciones en un cien por ciento, al final lo que se puede hacer es analizar la serie temporal más antigua que se tenga registro de cada acción con el intervalo más pequeño entre cada precio para estudiar el comportamiento y así tomar una decisión financiera, entre más información se tenga para analizar mejor precisión en los cálculos.

La parte complicada es decidir de forma óptima una operación, monitorear las actividades activas, aplicar las medidas de riesgo adecuadas, como órdenes de *stop loss* y *take profit*, etc. Otro punto importante es descubrir ineficiencias estadísticas al adivinar la línea futura de la Bolsa.

Bibliografía

- [1] H. Markowitz, *Portfolio Selection: Efficient Diversification of Investments*, Yale: BlackWell Publishing, 1959.
- [2] W. JB, *The Theory of Investment Value*, Amsterdam: North Holland Publishing Company, 1964.
- [3] R. Lawrence, *Using Neural Networks to Forecast Stock Market Prices*, Manitoba: Department of Computer Science, 1997.
- [4] J. Del Carpio, "Las Redes Neuronales Artificiales en las Finanzas," *Revista de la Facultad de Ingeniería Industrial*, vol. 8, no. 2, pp. 28-32, 2005.
- [5] A. Álvarez and A. Trillos, *Apoyo a la investigación: Estructuración de Portafolios de Inversión en Acciones Usando Redes Neuronales; Uso de los Métodos Tradicionales*, Bucaramanga: UNAB, 2005.
- [6] O. García and A. Morales, "Las Redes Neuronales Artificiales como una herramienta de Análisis en la determinación de las Empresas que Permanecen Listadas o Deslistadas dentro de la Bolsa Mexicana de Valores," *UAEM, UNAM, CIENCIA@UAQ*, vol. 5, no. 2, p. todas, 2012.
- [7] E. Carmona, "Ajuste a la Calificación del Riesgo de mercado de las Acciones más Volátiles que Conforman el Índice de Precios y Cotizaciones de la Bolsa Mexicana de Valores," *Revista Mexicana de Economía y Finanzas*, vol. 8, no. 1, pp. 25-51, 2013.
- [8] J. Frausto, E. Álvarez, J. Sánchez, P. Limonchi and J. Lebrun, *Análisis de Portafolios de Inversión para la Bolsa Mexicana de Valores*, Spain: Economía Digital, Temas Selectos de Economía, 2015.
- [9] N. Mancilla , M. Muñoz and M. Sánchez, *Estructuración de portafolios mediante el uso de redes neuronales: un comparativo de la rentabilidad frente a los métodos tradicionales*, Bucaramanga : UNAB, 2019.
- [10] Y. Hilpisch, *Artificial Intelligence in Finance, A Python-Based Guide*, Sebastopol: O'Reilly, 2020.
- [11] X. Brun and M. Moreno, *Análisis y Selección de inversiones en mercados financieros*, España: PROFIT, 2008.
- [12] R. Stephen, W. Randolph and J. Branford, *Fundamentos de Finanzas Corporativas*, Ciudad de México: Mc Graw-Hill, 2014.
- [13] P. Domingos, *The Master Algorithm : How the Quest for the Ultimate Learning Machine Will Remake Our World*, New York: Basic Books, 2015.
- [14] Goodfellow, I. Y. Bengio and A. Courville, *Deep Learning*, Cambridge: MIT Press, 2016.
- [15] M. and T. M, *Machine Learning*, New York: McGraw-Hill, 1997.
- [16] H. Yuval Noah, *A Brief History of Tomorrow*, London: Harvill Secker, 2015.
- [17] Chollet and Francois, *Deep learning with Python*, Shelter Island: Manning, 2017.
- [18] J. Vander Plas, *Python Data Science Handbook*, Sebastopol: O'Reilly, 2017.
- [19] Agrawal, Ajay, G. Joshua and G. Avi, *Prediction Machines: The Simple Economics of Artificial Intelligence*, Boston: Harvard Business Review Press, 2018.
- [20] Copeland, Thomas, F. Weston and K. Shastri, *Financial Theory and Corporate Policy*, Boston:

Pearson, 2005.

- [21] E. F. Fama, "Random Walks in Stock Market Prices," *Financial Analysts Journal*, vol. 21, no. 5, pp. 55-59, 1965.
- [22] P. Norvig, F. Pereira and A. Halevy, "The Unreasonable Effectiveness of Data," *IEEE Intelligent Systems, Expert Opinion*, p. 12, 1 March/April 2009.
- [23] Y. Hilpisch, *Python for Finance: Mastering Data-Driven Finance*, Sebastopol: O'Reilly, 2018.
- [24] M. Jensen, "Some Anomalous Evidence Regarding market Efficiency," *Journal of Financial Economics*, vol. 6, no. 2/3, pp. 95-101, 1978.
- [25] M. Tegmark, *Life 3.0. Being Human in the Age of Artificial Intelligence*, United Kingdom: Penguin Random House, 2017.
- [26] R. S. Tsay, *Analysis of Financial Time Series*, Hoboken: Wiley, 2005.