

Grau en Estadística

Títol: Implementació de bot de trading realitzant prediccions mitjançant sèries temporals i optimització amb IA

Autor: Roger Garcia Marfany

Director: Dr. Ramón Alemany i Dr. Ernest Pons

Departament: Departament d'Econometria, estadística i economia aplicada

Convocatòria: 2n SEMESTRE curs 2021



Resum

El desenvolupament de noves tecnologies creix a un ritme exponencial, els casos en els quals un o varis ordinadors, poden substituir el treball humà són cada cop més freqüents. Històricament els oficis que les màquines podien substituir a l'humà es basaven únicament en tasques manuals i que col·loquialment parlant, requereixen poc recurs intel·lectual. L'ús de softwares automàtics per replicar càlculs complexos s'apliquen en in comptables sectors. Els bots de *trading* son un clar exemple de software que pot substituir el treball humà més manual i de càlcul per a poder automatitzar processos estratègics per a prendre decisions. En l'actualitat existeixen molts bots de *trading* operatius, reservats a un públic molt selecte i operant de forma privada. El *trading* ja de per si és una activitat molt arriscada, i si es barreja amb el món dels criptoactius parlar de risc és un eufemisme de la realitat. Els criptoactius sorgeixen fa 10 anys de mà de Satoshi Nakamoto, un geni de la criptografia que desenvolupa un mètode de pagament descentralitzat que més endavant resulta en el mercat més especulatiu de la historia. La implementació d'un bot en aquests mercats pot donar vertigen, però te tota una sèrie d'avantatges a les que cal parar l'atenció. El fàcil accés i el no requeriment d'un dipòsit ni volums mínims d'operació sumat a l'horari de negociació sense *stop*, permeten a qualsevol *trader* tant expert com amateur operar a qualsevol hora del dia i des de qualsevol dispositiu sense requerir grans capitals.

Keywords- Bitcoin, trading, exchange, bot, time series, machine learning, dynamic models, data analysis , predictions, permutation tests.

Abstract

The development of new technologies is growing at an exponential rate, cases in which one or more computers can replace human labor are becoming more common. Historically, the activities in which machines could replace humans were based only on manual tasks, and in general, activities that require little intellectual effort. Nowadays, the use of automated software to replicate complex calculations is applied in countless sectors. Trading bots are a clear example of software that can replace manual and computational human labor in order to automate strategic decision-making processes. There are currently many trading bots operating, reserved for a very select audience and operating privately. Trading is already a very risky activity, and if mixed with the world of cryptocurrencies talking about risk is a euphemism for reality. Cryptocurrencies emerged 10 years ago from the hand of Satoshi Nakamoto, a cryptography genius who developed a decentralized payment method that later resulted in the most speculative market in history. Implementing a bot in these markets can be mind-boggling, but it has a number of advantages that need to be addressed. The easy access and the non-requirement of a deposit or a minimum trading volume added to the non-stop trading hours, allow any expert or amateur trader to trade at any time of the day and from any device without requiring a large capital.

Keywords- Bitcoin, trading, exchange, bot, time series, machine learning, dynamic models, data analysis, predictions, permutation tests.

CLASSIFICACIÓ AMS

37M10 - Time series analysis

93A30 - Mathematical modeling (models of systems, model-matching, etc.)

47N30 - Applications in probability theory and statistics

62F40 - Bootstrap, jackknife and other resampling methods

92B20 - Neural networks, artificial life and related topics

97R50 - Data bases, information systems

Agraïments

Volia agrair en primer lloc als meus pares per sempre haver estat al meu costat, fer-me veure que l'esforç porta recompensa i que val més arriscar-se que penedir-se.

També m'agradaria agrair als meus tutors Ramón Alemany i Ernest Pons, per accedir a guiar-me durant tot aquest camí i per tota l'ajuda rebuda.

M'agradaria agrair a tots els meus companys de carrera per aquests últims 4 anys, però sobretot, a Xavier Ventayol, Xavier Martínez, Arnau Reynals, Carlota Ruiz de Porras i Laura Salvadó per haver fet de l'experiència universitària un procés molt agradable i que tornaria a repetir sense dubte.

Finalment, agrair també a tots els companys i família que, tot i haver escollit un tema per al TFG una mica rebuscat, m'han donat suport a i animat a seguir endavant.

Índex de continguts

I Resum

II Abstract

1. INTRODUCCIÓ	7
1.1. Motivació	7
1.2. Bitcoin	8
1.3. Bot de trading	11
2. OBJECTIUS	13
3. DESENVOLUPAMENTS PRELIMINARS	15
3.1. Enllaç amb el <i>Broker</i>	15
3.2. Modificació de la llibreria Github	16
3.3. <i>Webscrapping</i>	19
4. MARC TEÒRIC	22
4.1. Models <i>ARIMA</i>	22
4.2. <i>Multilayer Perceptron (MLP)</i>	24
4.3. <i>Extreme learning machines (ELM)</i>	26
5. ELABORACIÓ PRELIMINAR DEL BOT	27
5.1. Definicions i funcions prèvies	27
5.2. Validació i evaluació	29
6. RESULTATS I DISCUSSIÓ	33
6.1. Programa Backtesting	33
6.2. Valor acumulat en Cartera	38
6.2.1. <i>Percentatge d'encerts</i>	38
6.2.2. <i>Valor portafoli interès compost</i>	39
6.3. Interacció entre els mètodes	40
6.3.1. <i>Correlació en valor de cartera acumulat</i>	40
6.3.2. <i>Estandardització</i>	43

6.4. Anàlisi rendibilitats simples	46
6.4.1. <i>Comparació rendibilitats simples amb compostes</i>	47
7. CONCLUSIONS	49
8. BIBLIOGRAFIA	52
9. ANNEX	55

1. INTRODUCCIÓ

1.1. Motivació

Un bot de *trading* algorítmic és un *software* que genera senyals de compra i venda per un determinat grup d'actius per tal d'esbrinar en quins moments és millor realitzar segons quin tipus d'operativa. La majoria de bots de *trading* que hi ha actualment en el mercat reproduïxen l'anàlisi tècnica dels actius de forma que el *software* que hagin generat pugui decidir de forma autònoma i, sota certs valors d'uns paràmetres predefinitos, quan és un bon moment per a comprar o vendre accions. En el marc d'aquest projecte, l'actiu escollit ha estat la criptomoneda Bitcoin. La motivació principal per a escollir aquest tema ha estat més un repte intel·lectual que una qüestió conceptual. Aquest últim any 2020-2021 ha vist un auge de les criptomonedes. Moltes firmes d'inversió han intentat predir si aquesta potencial bombolla¹ del Bitcoin acabaria desinflat-se o implosionant, i fins a quin preu podia arribar la cotització d'aquesta criptomoneda. Una altre raó de pes per escollir aquest actiu ha sigut l'horari de negociació que té envers a l'euro, que és de 24 hores del dia els 365 dies de l'any, motiu pel qual a l'hora de calcular prediccions sobre el preu, gairebé es podrien considerar les dades com a contínues. Addicionalment, a l'hora de fer comprovacions i implementar l'anàlisi, no s'ha depès del típic horari de negociació que tenen les accions a la borsa, que només és de 9:00 a 17:30 en cada franja horària.

La idea d'elaborar un bot de *trading* i la motivació emergeixen del repte personal de poder crear un producte que generi un valor residual econòmic. Tot i ser una idea molt ambiciosa per a un estudiant de grau, i que potser no es pot assolir l'objectiu, és un projecte que sempre es podria reprendre i provar noves metodologies. La majoria de bots de *trading* que es poden comprar al mercat funcionen de forma molt senzilla i es guien, tant per baixades percentuals de preus a l'hora de posar *limit orders*², com per indicadors tècnics com el *RSI*³(*relative strength index*), *bandes de bollinger*⁴ i altres, que són indicadors més acadèmics que realment efectius en la predicció. En aquest treball, els mètodes d'elecció de moments i tipus d'operació s'obtidran després de seguir una estratègia de predicció de preus mitjançant models

¹ Es parla de bombolla especulativa en fases particulars del mercat en les que s'observa un augment considerable d'un preu o d'un producte. Un clar cas de bombolla que va afectar a Espanya va ser la bombolla immobiliària.

² Una *límit order* és una ordre de compra o venda a un preu específic. Si és una ordre de compra, només es pot executar pel preu fixat o menor. Si es una ordre de venda, només es pot executar al preu límit o superior.

³ L'*RSI* o índex de força relativa, és un indicador que mostra la força del preu mitjançant una comparació de moviments a l'alça o baixa dels preus de tancament. Es utilitza pels traders per mesurar la força d'una tendència i identificar senyals de fi de tendència.

⁴ Les bandes de *Bollinger* són uns indicadors usats en l'anàlisi tècnic dels mercats financers. Si els preus sobrepassen les bandes implica que el mercat està sobre comprat, si el preu sobrepassa per la banda inferior es considera sobre venut.

matemàtics. Es compararan tots els models ARIMA possibles amb combinacions de valors de p i q entre l'1 i el 5, i dos mètodes de *Machine learning* de xarxes neuronals, els *MLP*, *Multilayer perceptrons* i els *Extreme Learning Machine*. Aquests tres mètodes que s'expliquen més endavant, són models de predicció de sèries temporals i s'operaran de forma univariant, per tant, només és prendrà com a variable el valor de la cotització de tancament de cada hora.

1.2. Bitcoin

El Bitcoin és una nova classe de “diners” que es basa en una innovadora xarxa de pagaments. Aquest actiu utilitza una tecnologia *peer-to-peer*⁵ per operar sense cap autoritat central, és a dir, fora del control dels bancs i/o governs. La gestió de les transaccions i l'emissió de Bitcoins es duu a terme mitjançant un mètode col·laboratiu, de software lliure i de disseny públic en què tot el món hi pot participar. És, doncs, un protocol de codi obert i una xarxa entre iguals, una moneda a priori completament descentralitzada, és a dir, no existeix cap autoritat responsable de la seva emissió ni registre dels seus moviments. El Bitcoin opera sobre la *blockchain*, una tecnologia disruptiva que actua com llibre comptable on es registra cadascuna de les transaccions de forma “inalterable” (Segendorf, B., 2014).

Per tal d'entendre que és la *blockchain* i com funciona, cal definir uns quants conceptes.

Funció Hash: Aquesta funció és un algoritme matemàtic que transforma un input determinant en una sèrie de caràcters de sortida. Bàsicament, s'utilitza per a protegir les contrasenyes i no guardar-les en una base de dades emprant un text clar i fàcilment intel·ligible. Un exemple molt senzill per a il·lustrar com funciona seria el següent: suposem que convertim cada lletra d'una paraula en el número de la posició que ocupa a l'abecedari. Per exemple, la paraula gos es convertiria en: 71519 (7=G) (15=O) (19=S). Vist aquest exemple és molt senzill saber com convertir qualsevol paraula aplicant la funció *Hash*. El punt complicat ve quan coneixem el *Hash* però no el procés mitjançant el qual s'ha convertit l'input en l'output. En concret, el Bitcoin empra el *SHA-256*, que pertany a la família *MD* de les funcions *Hash* (Gilbert, et al., 2003). Es pot exemplificar amb la imatge presentada a la *Figura 1.2.1*:

Input	Output (Hash)
Cat	93g56gtf229hbno00r45sktrpbs59so9r3t7saer
A white cat is outside	js03bbstgo94r6s1z8mg05fgt3sba9tob32bsap7
A white cat is inside	bbr19007go2tsi52bsi50o21nmiseas45on23mjn
A whiet cat is inside	339n5sbk249nb9530gjdI04h92jg02jg9sm93hpsz
A white cat is insid	4bbj390osoh9djm395bksh94gf03sg034dfjh31x

Figura 1.2.1: Exemple d'Output Hash sobre un Input que mostra una frase amb molt poca variació

⁵ Una xarxa *peer-to-peer* (*P2P*) és una xarxa d'ordinadors en la que tots o la majoria d'aspectes funcionen sense clients ni servidors fixes, sinó amb una sèrie de nodes que es comporten igual entre si. Les xarxes *P2P* permeten l'intercanvi d'informació en qualsevol format entre ordenadors interconnectats

S'observa que cap de les sortides té res a veure amb les entrades, el procés mitjançant el qual es generen aquests outputs comporta que sigui extremadament complex endevinar com revertir-los. Un petit canvi en qualsevol caràcter i provoca que el *Hash* sigui completament diferent.

Nonce (number that can only be used once): Aquest terme denomina un número aleatori i de característiques úniques. En la xarxa de Bitcoin és emprat per generar un bloc *ID* (seguretat extra per la *blockchain*).

Nodes: Els nodes són els ordinadors que componen la xarxa i que la validen. Cada vegada que es produeix una transacció s'ha d'aprovar per tots els nodes. Cada cop que un node ha verificat una transacció, hi ha un vot electrònic on els nodes decideixen si les transaccions són vàlides o no. En cas que decideixin que sí, s'escriu en el bloc. L'avantatge és que cada ordinador té una còpia del llibre de registres a la *blockchain* i, per aquesta raó, falsificar qualsevol transacció implicaria modificar-la en tots els ordinadors del món i la potència de computació que és necessària seria de més del 50% de la que s'utilitza per minar (definit més endavant).

Bloc: És el nom que reben el conjunt de transaccions confirmades i la informació addicional que s'ha unit a la cadena. Cada bloc que en forma part està compostat per un codi alfanumèric que enllaça cada bloc amb l'anterior, conté el paquet de transaccions que inclou i que vol codificar, i un altre codi alfanumèric que enllaça amb el següent bloc (*Figura 1.2.2*).

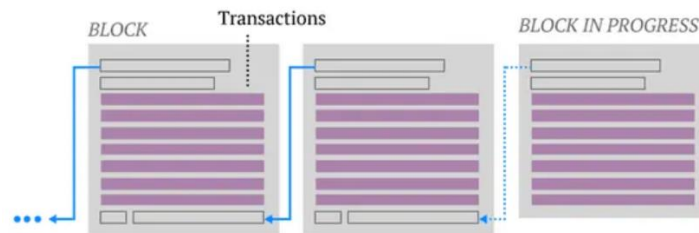


Figura 1.2.2. Esquema de funcionament de la blockchain

Mineria: Procés mitjançant el qual es resolen problemes matemàtics usant equips informàtics. Els problemes a resoldre bàsicament són descobrir el *hash*. És a dir a partir de l'*output* trobar l'*input* que el genera. La forma de trobar-lo és, amb potència bruta, generar totes les possibles combinacions d'*output* i anar "preguntant" si s'ha encertat l'*input*. El primer de tots els ordinadors que trobi la combinació correcta que encaixi amb la solució plantejada pel bloc rebrà una recompensa. Actualment, per al minat del Bitcoin, el sistema de consens utilitzat és el *Proof-of-Work (PoW)* ⁶ un protocol segur, senzill d'implementar algorímicament i amb una

⁶ El *Proof-of-Work* serveix per evitar comportaments indesitjats en una xarxa. Funciona sota el concepte de requerir treball al client que posteriorment es verificat per la xarxa. En el cas del Bitcoin consisteix en la resolució d'operacions de càlcul complexes. Un exemple de *PoW* seria el *captcha* d'algunes pàgines per verificar que no som un robot.

gran capacitat de resistència a atacs, però que, malauradament, consumeix una gran quantitat d'energia elèctrica.

Bitlletes Una bitlleta és una cadena de números i lletres com el següent: 18c177926650e5550973303c300e136f22673b74. És una direcció que apareix en varis blocs dins de la *blockchain* quan es realitzen transaccions. És l'identificador d'una persona. Sempre es podrà consultar quines transaccions ha fet, però mai la seva identitat. La direcció de cada bitlleta és, doncs, una clau pública⁷.

Tot i que el Bitcoin és una criptomoneda envoltada per una forta especulació, presenta un sistema d'independència econòmica de qualsevol institució. És un actiu deflacionari, atès que la recompensa dels miners per validar cada bloc es veu reduïda a la meitat cada 4 anys i el nombre màxim de Bitcoins que existiran en circulació està predifinit i és de 21 milions. Presenta una tecnologia molt segura a dia d'avui que, a priori, ni la supercomputació quàntica podria desestabilitzar. Quant major sigui el nombre d'usuaris i de miners, major la seguretat de la xarxa. Tot i estar precedida d'unes quantes bombolles, la realitat és que, a dia d'avui, el Bitcoin ja és un actiu real amb una capitalització total de mercat de 577.293.037.903 € (22/05/2021). Per posar-ho en perspectiva, aquesta quantitat equival al voltant del 50% del PIB anual a Espanya del 2020 (1.121.698 M€). En certes economies fortament afectades per la inflació i per governs totalitaris, el Bitcoin pot servir de valor refugi⁸ i de via de llibertat, atès que no es poden evitar les transaccions de persona a persona ni tampoc l'acumulació de criptomonedes. De fet països com El Salvador, ja la reconeixen com a moneda oficial del país. (BBC News., 2021)

En general, les avantatges que representa Bitcoin enfront d'altres monedes són:

- **Rapidesa:** Enviament de qualsevol quantitat de diners en qüestió de minuts sense importar ni la quantitat ni el destí, i sense que cap govern o institució pugui evitar-la.
- **Global:** Es poden enviar Bitcoins a qualsevol país del món de la mateixa forma que un e-mail, no té fronteres. És la primera moneda que garanteix la llibertat operativa absoluta dels seus usuaris.
- **Emissió descentralitzada:** Cap govern ni banc central pot controlar ni interferir en la valoració, ni en la creació, ni en la seva distribució.
- **Transparència:** A causa de la naturalesa de la tecnologia que el suporta, totes les transaccions són públiques sota pseudònim i per tant, es pot rastrejar qualsevol

⁷ Una clau pública és un identificador que es pot compartir a voluntat i que permet interactuar amb altres usuaris. És una de les dos parts que formen el conjunt de claus creades per la criptografia asimètrica per compartir informació de forma segura

⁸ Un valor refugi és un actiu amb menys volatilitat que altres que acostuma a ser mantingut en temps d'incertesa econòmica quan els mercats es contrauen. Històricament el valor refugi més negociat ha estat l'or, molt útil per a mantenir poder adquisitiu en períodes d'alta inflació monetària.

pagament. Actualment, ja no s'utilitza tant en la *Deep web*⁹, justament per la seva traçabilitat.

- Codi obert: és un *software* obert, lliure i totalment gratuït. Qualsevol persona pot estudiar-lo o intervenir en millorar-lo.
- Basat en el consens, d'emissió limitada i sense barreres.

Els desavantatges principals i els reptes actuals als que s'enfronta el Bitcoin per a poder ser un mètode de pagament efectiu són:

- Les transaccions són irreversibles, per tant, un error humà el paga el propi usuari.
- El valor del Bitcoin està totalment sotmès a la demanda del mercat, per tant, no serveix com a mètode de pagament habitual a causa de la gran volatilitat que l'envolta. Tanmateix, queda a anys llum de la inflació de països com Argentina o Veneçuela.
- Molts comerços encara no hi confien, ja que es té la idea que és una moneda utilitzada per delinqüents, per provocar fraus i per especular. Sovint, s'obliden que, per exemple, el 90% dels bitllets de dòlar als EEUU presenten traces de cocaïna (López., 2009)

1.3. Bot de *trading*

Un bot de *trading* és un *software* que permet la compra-venda d'actius financers de forma automàtica o semi-automàtica.

L'avantatge d'aquests programaris (*softwares*) és que permeten l'operació al mercat a qualsevol hora i moment, tot seguint uns patrons prèviament definits. Bàsicament, “substitueixen” la figura del inversor per la d'un programa d'ordinador. Els inversors manuals sovint no poden reaccionar amb la suficient rapidesa per aprofitar totes les oportunitats que apareixen en les sessions. A part, la fatiga i cansament després d'hores i hores d'anàlisi gràfic poden portar als inversors manuals a prendre males decisions. Amb els bots, tots aquests problemes desapareixen, atès que l'ordinador no es cansa i sempre serà més ràpid en càlcul que l'humà.

La majoria d'aquests bots funcionen amb algorismes senzills. No obstant, certes entitats inverteixen molts milions d'euros en el desenvolupament de bots que tinguin altes taxes de retorns. Els bots de *trading* als que pot tenir accés un ciutadà comú no acostumen a funcionar massa bé. Per això, bancs i grans entitats financeres inverteixen milions d'euros per a poder desenvolupar-ne alguns que siguin rentables i factibles. Certs bots propietat de grans corporacions financeres analitzen moltes variables, tant econòmiques com gràfiques, dels actius amb els que pretenen generar guanys operant. En el cas d'aquest treball, el primer enfoc

⁹ La *Deep web* està formada per tot aquell contingut no indexat a motors de cerca convencionals. És el internet invisible en el que abunden hackers, traficants, i agents del FBI. En alguns països del món és il·legal accedir-hi, és el internet sense censura on l'anonimat és un jugador estrella. El Bitcoin degut a la seva anonimitat i mètode de pagament digital, va ser una de les primeres formes de pagament que es van utilitzar a la *Deep web*.

serà univariant, i es prendrà com a variable la cotització del Bitcoin. Es considerarà com un èxit a priori, el cas en que el % de prediccions totals efectives sigui major al 50%. Un cop obtinguda una gran base de dades amb casos reals de prediccions, es buscarà optimitzar a posteriori en quins casos “s’hauria” de fer cas a les prediccions i en quins casos no. També es calcularà la posició del valor de la cartera obtinguda en els casos en els que s’hagués operat. En última instància, i si les conclusions del treball són positives, s’automatitzarà el bot per a que pugui comprar i vendre de forma automàtica sense haver de passar per cap humà.

2. OBJECTIUS

Els treball té tres objectius principals:

1. Creació d'un *bot de trading* mitjançant el qual es pugui obtenir una certa rendibilitat en el mercat de les criptomonedes.
2. Execució del programa, de forma que el bot pugui de forma autònoma realitzar l'operació.
3. Validació del bot i comprovació del seu funcionament, mitjançant la creació d'un programa de *backtesting*¹⁰.

Per tal de desenvolupar els tres objectius:

1. En primer lloc, es crearà un software mitjançant el qual es pugui obtenir una certa rendibilitat en el mercat de les criptomonedes, concretament amb el *pair*¹¹ *Bitcoin-euro*. El bot constarà de tres parts, la primera realitzarà una lectura de dades (en temps real) per obtenir les cotitzacions horàries en cada moment del dia emprant *webscrapping*¹². La segona consistirà en l'avaluació d'aquestes dades per a decidir que fer en aquell mateix moment de temps, és a dir, decidir si emetre una ordre de compra o una de venda. Les ordres són de naturalesa binària, COMPRAR si es creu que l'actiu pujarà d'ara fins al següent moment de temps o VENDRE si es creu que l'actiu baixarà des d'un moment de temps concret fins al pròxim. La decisió de comprar o vendre vindrà determinada per un dels tres mètodes que s'exposen més endavant. Aquests tres mètodes són models predictius de sèries temporals.
2. Un cop es decideixi si l'actiu pujarà o no, el bot comunicarà quina és l'acció a realitzar de tal manera que el propi bot pugui de forma autònoma realitzar la operació. Per tant, la tercera funció del bot serà executar les ordres amb diners reals. Aquesta part del treball es realitzarà només en cas que es pugui concloure el bot és consistent i permet obtenir uns guanys assegurats. En un futur es podria considerar continuar optimitzant el bot per maximitzar guanys o simplement per no obtenir pèrdues en el cas de que el bot no acabi de funcionar bé.

¹⁰ El *backtesting* es un procés de testatge d'una estratègia abans de posar-la en pràctica. En el cas particular del *trading*, permet a l'usuari conèixer si les prediccions i l'estratègia escollida són correctes, i quins resultats s'haguessin obtingut operant en diners reals.

¹¹ El *webscrapping* és una tècnica utilitzada mitjançant programes de software per obtenir informació de llocs web. Usualment aquests programes simulen la navegació d'un humà en la *World Wide Web (WWW)*. És el procés que permetrà obtenir les dades dels preus de cotitzacions del Bitcoin a l'script d'R.

¹² Un *pair* o parell, representa la relació biunívoca entre dos objectes negociables. En aquest cas representa el preu de Bitcoin en euros. Els *tickers* (símbols que representen actius) del parell que es negocia són. BTC (Bitcoin) i EUR (euro).

3. Per tal de validar si el bot funciona bé o no abans d'arriscar diners reals, posteriorment a la creació del bot, però anterior a la implementació d'aquest, es generarà un programa de *backtesting* per avaluar si el bot és efectiu o no, es compararan els mètodes per veure quin funciona millor en cada cas i es decidirà quina estratègia s'ha de seguir per optimitzar els guanys.

3. DESENVOLUPAMENTS PRELIMINARS

Per aclarir l'estructura, a la *Figura 3.1* es mostra l'esquema de com funcionarà el bot de *trading*. En blau s'expressen els processos que e seguiran i les fletxes indiquen la direccionalitat d'aquests. En taronja s'exposa que és el que s'està calculant o quina utilitat te cada transició.

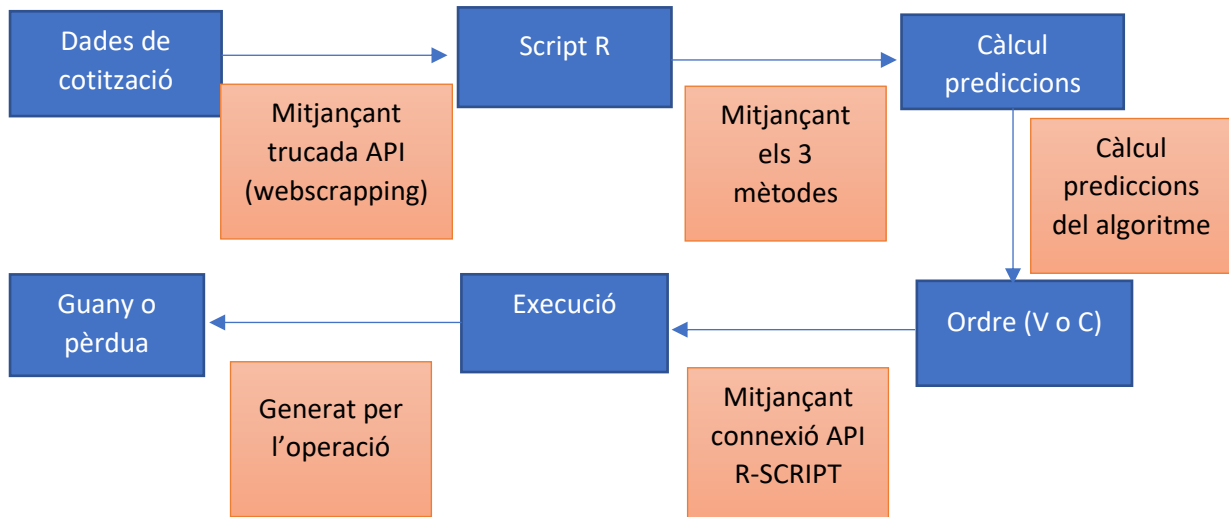


Figura 3.1. Esquema estructura bot de trading

Primer s'obtenen els valors de cotització del Bitcoin amb la periodicitat escollida (horària en aquest cas) mitjançant una trucada *API*¹³. Un cop obtingudes s'analitzen a l'R i es calculen les prediccions per als tres mètodes. S'utilitzarà com a vàlida aquella que en el moment es consideri segons uns paràmetres determinats i segons les conclusions que s'obtinguin en el desenvolupament del programa de *backtesting*. Un cop calculades les prediccions s'emetrà un ordre de compra o de venda (*límit order*) on el preu especificat serà el mateix al que ha tancat la última hora. Mitjançant de connexió *API* i enllaç amb la clau pública y privada es podrà enviar l'ordre al broker per a posteriorment ser executada. Després de ser executada es coneixerà si s'ha produït un guany o una pèrdua.

3.1. Enllaç amb el *Broker*

El primer pas per a construir un bot de *trading* és tenir un enllaç o *link* entre l'*script* des d'on es poden realitzar les operacions i l'*Exchange* per operar i efectuar-les. Què és l'*Exchange*? En el món de les criptomonedes, rep el nom d'*Exchange*, l'intercanvi de criptomonedes virtual, és un espai on es realitzen accions de compra-venda d'aquests actius. L'objectiu és simplement

¹³ Una trucada *API* (*Application Programming Interface*) és una instància en la que el propietari d'un lloc web sol·licita l'ús d'informació d'un altre. En el cas del treball permet enllaçar l'*script* d'R amb la pàgina web del broker i permet la interacció (comprar, vendre, obtenir preus, etc.) al *broker* des d'un *script* d'R.

permetre als usuaris (*traders*) participar en un mercat en el que poden obtenir guanys o pèrdues gràcies a les variacions dels preus, o simplement, acumular criptoactius. L'existència dels *Exchange* permet dinamitzar la vida econòmica i financera de les criptomonedes, atès que faciliten l'intercanvi de diner fiduciari¹⁴, per criptomonedes o viceversa, des de qualsevol part del món.

Com que en aquest projecte, totes les operacions es pretenen fer des de un *script* de l'R, s'ha escollit l'*exchange* Coinbasepro. El contacte entre *script* i *Exchange* s'ha realitzat per trucada *API*. El dipòsit inicial s'ha realitzat amb targeta de debit (Figura 3.1.1).

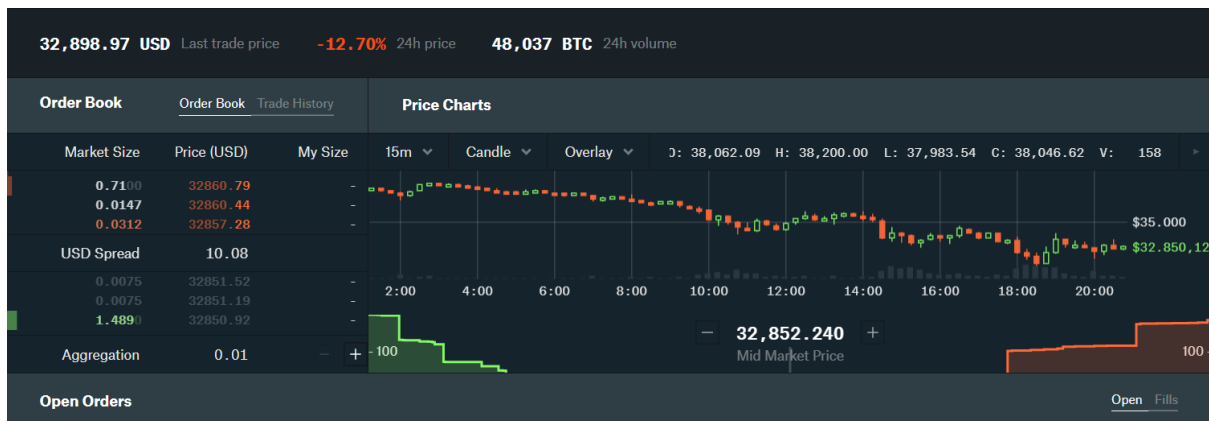


Figura 3.1.1. Captura de pantalla del bróker Coinbasepro

3.2. Modificació de la llibreria Github

La llibreria emprada per a poder realitzar aquesta connexió ha estat la llibreria *rgdax* de Dheeraj Agarwal. Un cop instal·lat el paquet des de *CRAN*¹⁵, es van comprovar les funcions predefinides.

La funció *account()* de la llibreria *rgdax* és la que permet comprovar el saldo al compte des de R. Es va utilitzar com a validació d'accés a *Exchange* des de R. Va sorgir un primer problema amb aquesta funció, ja que no permetia connectar l'*script* amb el *broker*. L'aparició d'errors estranys, de difícil adscripció fou molt freqüent (Figura 3.2.1). Navegant per *stackoverflow*, *github* i diversos fòrums, es va descobrir que hi havia un paquet de dependències que no s'executava bé amb R, a causa que *Rcurl*, un paquet de R, no era compatible amb els ports de connexió necessaris per a realitzar correctament el *webscrapping*.

¹⁴ El diner fiduciari o col·loquialment conegut com a diner *FIAT*, és un sistema monetari en el qual la moneda no es pot convertir en metalls preciosos i per tant, la seva circulació es basa en la confiança de l'institut emissor.

¹⁵ *CRAN* (*The Comprehensive R Archive Network*) es una xarxa de *ftp* (*file transfer protocol*) i servidors web al voltant del món que emmagatzemen informació idèntica actualitzada, versions de codi i documentació per a l'R. És la pàgina des d'on, per defecte, s'instal·len tots els paquets de funcions de l'R.

```
Fehler in function (type, msg, asError = TRUE) :  
error:1407742E:SSL routines:SSL23_GET_SERVER_HELLO:tlsv1 alert protocol version
```

Figura 3.2.1. Error observat per la funció `account` de la llibreria `rgdax`:

Aquest error era degut a que la llibreria `Rcurl` utilitzava seguretat `tsl1` (antiquada i en desús). Després d'indagar durant moltes hores per internet, cercant sense massa èxit, es va trobar que una de les possibles solucions a aquest problema era fer un *fork* de la llibreria i instal·lar-la des del meu propi repositori de github, eliminant la llibreria dependent `Rcurl` per la `hhtr` que opera en `tlsv1.2`¹⁶, i així evitar l'aparició de l'error (Figura 3.2.2).

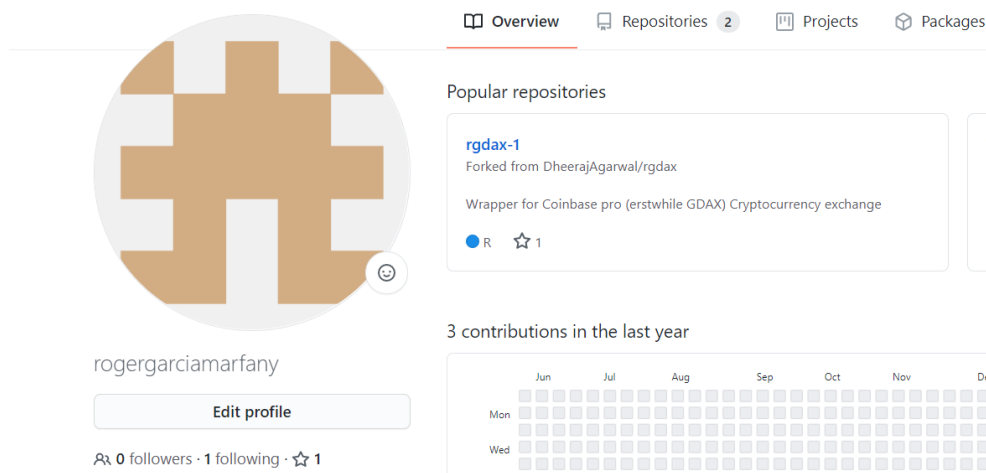


Figura 3.2.2. Captura de pantalla del compte propi de github

En aquells moments, es va comunicar amb el desenvolupador, Dheeraj Agarwal, per tal de comprovar si ell podia solucionar el problema detectat ja que, tot i haver trobat una solució provisional, era una mica rudimentària i, a fi de comptes, la llibreria era de creació seva. Es transcriu a la pàgina següent, part de la conversa mantinguda (Figura 3.2.3).

¹⁶ *TLS* (*transport layer security*) es un protocol criptogràfic que proporciona comunicacions segures per la xarxa d'internet. Concretament la versió `tlsv1.2` va ser llançada l'agost de 2009. L'any 2013 es va llançar l'última versió (`1.3`) i algunes pàgines d'internet van donar com obsoletes les versions anteriors degut a la major seguretat, rapidesa que aquesta nova oferia.

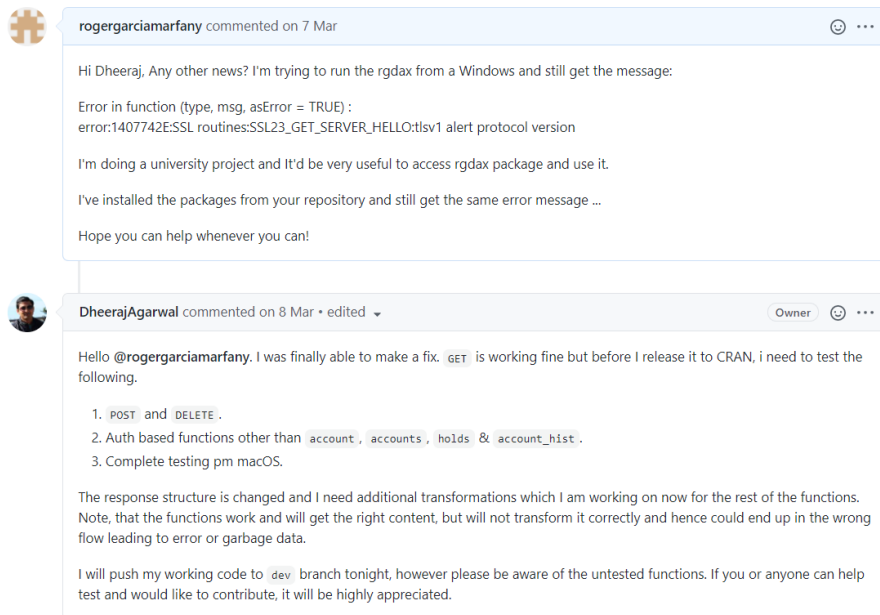


Figura 3.2.3. Captura de pantalla conversa amb developer de rgdax

Finalment el propi desenvolupador de la llibreria va modificar i actualitzar el paquet perquè no donés més errors a qualsevol persona que se l'instal·les canviant la desfasada llibreria *Rcurl* per la *httr* (Figura 3.2.4).

The screenshot shows a GitHub repository page for 'DheerajAgarwal working version with httr'. The repository has 81 commits and was last updated on 8 Mar. The file list includes folders for R, man, and tests, and various configuration and documentation files.

File/Folder	Description	Updated
R	working version with httr	3 months ago
man	working version with httr	3 months ago
tests	temp changes to auth using httr	3 years ago
.Rbuildignore	added configs for CRAN and Travis	4 years ago
.gitignore	ignoring the credential file	4 months ago
.travis.yml	added configs for CRAN and Travis	4 years ago
DESCRIPTION	Updates for version 1.0.1	4 months ago
LICENSE	updated license based on feedback from CRAN	3 years ago
NAMESPACE	Updates for version 1.0.1	4 months ago
NEWS.md	Version 1.0.1 updates.	4 months ago
README.md	updated function list	4 months ago
cran-comments.md	changed base url and related content	3 years ago
rgdax.Rproj	changed cran package url to canonical form	2 years ago

Figura 3.2.4. Captura de pantalla de totes les carpetes de rgdax

Un cop sentada la base per a poder connectar l'*script* de R amb l'*exchange*, ja podia iniciar l'obtenció de dades per a poder construir models i fer prediccions per a, posteriorment, operar amb R. La funció per a realitzar la trucada en un principi havia de ser la pròpia del paquet *rgdax*; no obstant, com que per assolir els objectius del treball calia obtenir dades horàries, es va crear una de nova.

3.3. Webscrapping

El primer pas del *webscrapping* és realitzar una trucada *API* en un moment t determinat del temps. Per a la validació s'escull un moment t i s'obtenen les dades referents als k dies horàries. A la *Figura 3.3.1* es mostra esquema visual de com funciona el *webscrapping*, a partir d'un programa, s'extreu la informació d'una web per tal de poder emmagatzemar les dades posteriorment.

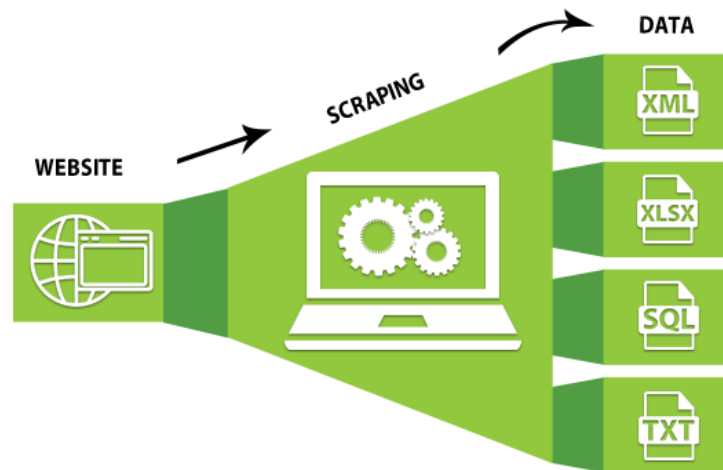


Figura 3.3.1. Esquema de webscrapping

La trucada *API* pren com arguments:

- **product_id** : “BTC-EUR” Fa referència al parell del qual es volen obtenir les dades, per aquest projecte només s’opera el Bitcoin però es podria modificar fàcilment per obtenir la informació sobre un altre actiu.
- **granularity**: Indica l’interval d’espaiament que hi ha entre observació i observació en la base de dades obtinguda per la trucada. L’escollida és 3600, que obté les dades d’hora en hora. (en 1 hora hi ha 3600 segons)
- **start_date, end_date**: Expressen la data d’inici i de finalització de la trucada d’*url*. Està especificada com la data actual (*Sys.time()* – 60 dies)
- **Variable trucada**: La trucada *query* obté en *OHLCV format*¹⁷.
- **Timestamp (V1)** Fa referència al moment t de temps de cada observació, degut a la *granularity* seleccionada cada observació està equiespaiada amb l’anterior i la següent per una hora.

¹⁷ *OHLCV*(*open-high-low-close—volume-chart*). És un format usat per obtenir precisament aquesta informació que compona el seu nom, els valors que s’expressen des de *V2* fins *V5*.

- **(V2): Open**, és el preu al que ha obert la cotització a l'interval escollit de temps. Per la granularity fixada serà el primer preu de cada hora.
- **H(V3): High**, és el preu màxim que hi ha hagut en l'interval de temps seleccionat que forma la vela¹⁸. És a dir, el màxim de cada interval horari.
- **L(V4): Low**, és el preu mínim que hi ha hagut en l'interval de temps seleccionat que forma la vela, és a dir, el mínim de cada interval horari.
- **C(V5): Close**, és el preu al que ha tancat la cotització en cada interval horari
- **V(V6): Volume**, és el volum que s'ha negociat en la moneda base seleccionada, és a dir en euros.

A les Figures 3.3.2 i 3.3.3 es mostren les gràfiques de les diferents variables que crida la funció.



Figura 3.3.2. Gràfic de les 3 primeres variables

¹⁸ Les veles japoneses en el *trading* són una eina que reflecteix en un gràfic informació sobre la cotització d'una acció en un moment determinat. Una vela és un boxplot que emmagatzema la informació del valor de la cotització. En un gràfic d'evolució de preu sencer, cada punt es pot representar com una vela, que serà vermella en cas de baixada en el període especificat per la *granularity* o verd en cas de pujada.



Figura 3.3.3. Gràfic de la última variable

Un cop obtingudes les dades s'haurà d'elaborar l'estratègia per a decidir si comprar o vendre en cada moment del temps. S'analitzaran tres metodologies que s'expliquen a continuació per a calcular prediccions i orientar les decisions de COMPRA i VENDA.

4. MARC TEÒRIC

4.1. Models *ARIMA*

Els models *ARIMA*, una metodologia economètrica basada en models dinàmics que utilitza dades de sèries temporals, va ser descrita per l'estadístic Box i l'enginyer Jenkins el 1970. Per poder identificar quin és el model *ARIMA* que millor explica com s'han generat un conjunt de sèries temporals és necessari que les dades siguin estacionàries. Una sèrie estacionària té un *shape* sense tendència (ni creixent ni decreixent, veure exemple a la *Figura 4.1.1*). Tampoc pot presentar fluctuacions de diferent amplitud. Aquesta sèrie s'obté mitjançant un procés de diferenciació. Es pretén trobar un model senzill de pocs paràmetres que permeti reproduir l'autocorrelació observada en les sèries temporals, amb una metodologia que funciona de mode iteratiu a l'hora de construir els models (De Arce,R, *et al*; 2003). L'operador retard, B és un operador tal que aplicat a una variable temporal la retarda;

$$By_t = y_{t-1}$$

La diferència regular és la que s'ha aplicat en tots els models ja que al comptar amb dades diàries i d'un període inferior a un any, no tenia sentit prendre diferències estacionals. Un operador tal que aplicat a una variable temporal la transforma de la següent manera:

$$\nabla y_t = (1-B)y_t = y_t - y_{t-1}$$

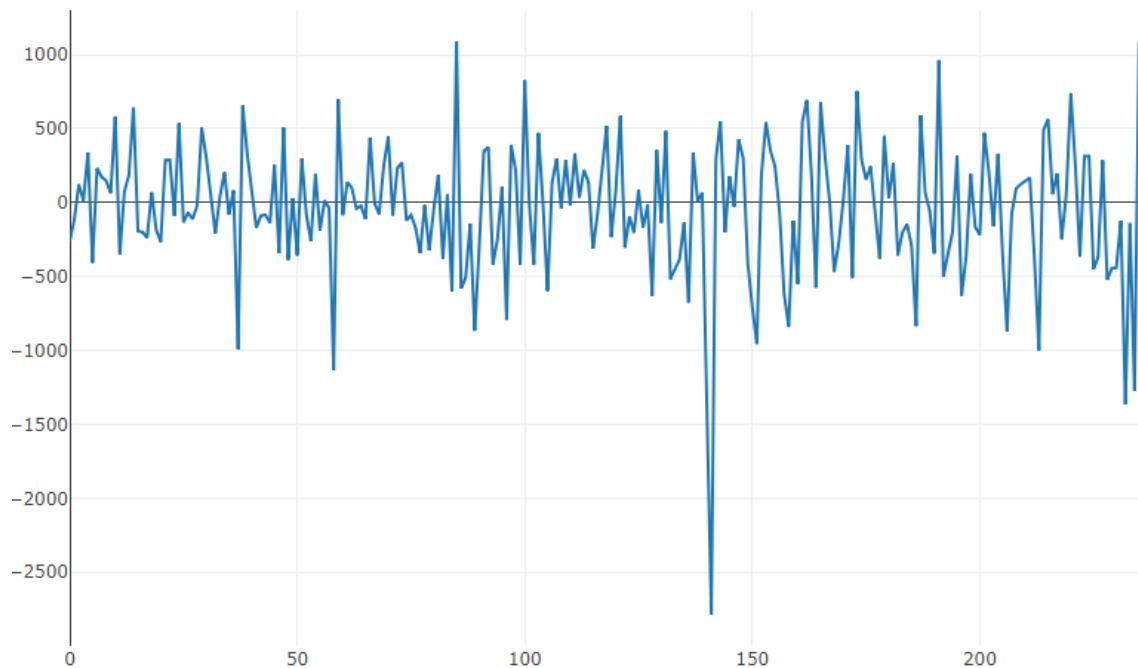


Figura 4.1.1. Exemple de serie estacionaritzada mitjançant diferències.(cotització de 10 dies horària).

Per tal de definir els models ARIMA és necessari definir be els següents conceptes bàsics:

- **Procés estocàstic** És una successió de variables aleatòries indicades segons el temps i definides en un espai mostral Ω . Cada una d'aquestes variables aleatòries te la seva pròpia funció de probabilitat i no te perquè estar correlacionada amb les altres. El conjunt de variables i cada variable constitueixen un procés estocàstic. En definitiva, es pot entendre com una família uniparamètrica de variables aleatòries indexades mitjançant el temps t .
- **Soroll blanc:** S'anomenarà soroll blanc a aquell procés estocàstic que compleixi:

$$E(\varepsilon_t) = 0 \quad \forall t \in Z$$

$$\text{var}(\varepsilon_t) = \sigma_\varepsilon^2 \quad \forall t \in Z$$

$$\text{cov}(\varepsilon_s, \varepsilon_t) = 0 \quad \forall s \neq t \in Z$$

- **Camí aleatori:** S'anomenarà camí aleatori a aquell procés estocàstic que compleixi:

$$y_t - y_{t-1} = \varepsilon_t$$

Els models ARIMA consten de tres parts associades a tres paràmetres diferents, p , d i q .

- **Ar(q) Models autoregressius.** Descriuen un procés en el que les observacions en cada moment del temps es poden predir emprant les observacions prèvies afegint-li un terme d'error aleatori. Aquest sempre és invertible.

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t$$

- **Ma(q), model de mitjanes mòbils.** En aquest tipus de model, el valor actual es prediu a partir dels valors de les variables aleatòries anteriors. Depenent del valor de q . Són processos que sempre són estacionaris.

Per tal d'identificar correctament els valors p i q , i conèixer quin és el model que millor explica una sèrie temporal, s'empren la *FAS* i la *FAP*, sigles que corresponen a funció simple d'autocorrelació simple i funció simple d'autocorrelació parcial, respectivament. El mètode convencional consisteix en observar ambdós gràfics per tal d'inferir quins podrien ser els valors p i q segons el nombre de coeficients significatius i el patró d'aquests gràfics. A les *Figures 4.1.2* i *4.1.3* es mostren les sèries del Bitcoin per als últims 10 dies.

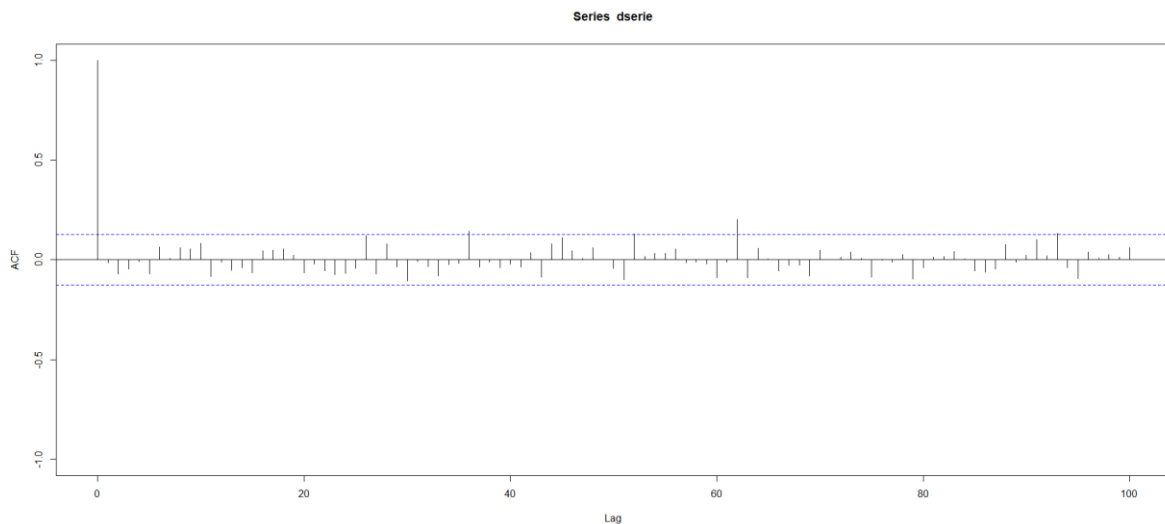


Figura 4.1.2. Gràfic ACF de la sèrie del Bitcoin:

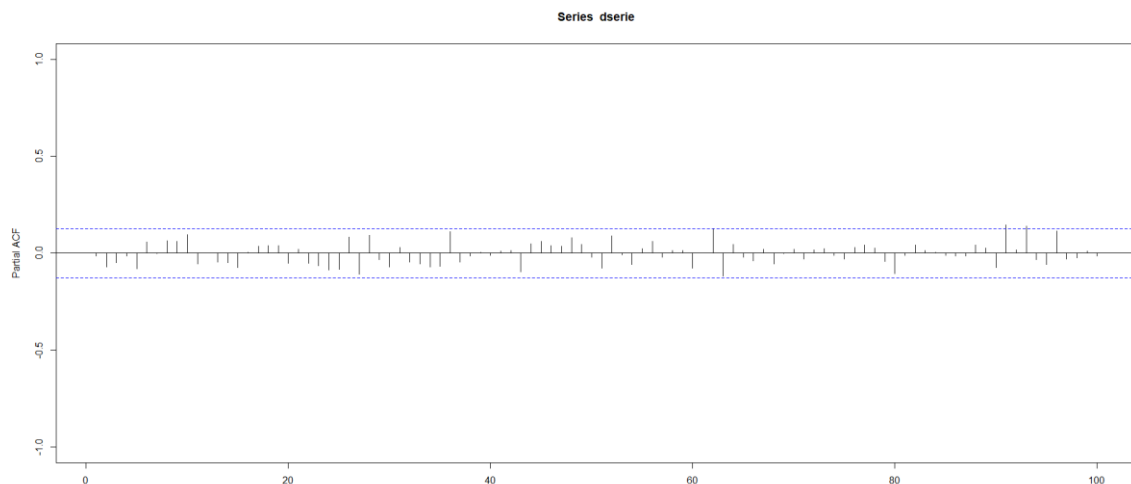


Figura 4.1.3. Gràfic PACF de la sèrie del Bitcoin:

Com que realitzar la lectura de forma automàtica d'ambdós gràfics cada hora seria impossible, en el treball, es seleccionarà com a millor model aquell que tingui un valor menor de *EP* (*error percentual*) d'aquells que encertin en la tendència. Aquest, serà utilitzat per a predir l'observació del següent període.

4.2. Multilayer Perceptron (MLP)

Un perceptró és una eina útil per a classificar conjunts de dades separables linealment. Aquest tipus de models es basen en xarxes neuronals. Una xarxa neuronal és un sistema computacional inspirat en com funcionen les neurones en un cervell animal. Aquests sistemes estan formats per unitats bàsiques, anomenades neurones, que es connecten entre si formant una xarxa. Aquestes xarxes, poden aprendre (construir interaccions) de tres formes: de forma supervisada, no-supervisada i reforçada. Les xarxes supervisades tindrien etiquetes que marcarien les dades

per tal d'ajudar a l'algoritme a validar-les a l'hora d'assignar *outputs*. En les xarxes no-supervisades, el propi algoritme troba les variants de patrons i correlacions. La xarxa reforçada es basa en la prova i error.



Figura 4.2.1. Exemple funció separable / no separable

No obstant, si s'intenta separar conjunts de dades que no s'ajusten a aquest patró com, per exemple, amb la funció *XOR*¹⁹, on el conjunt no és linealment separable, el *multilayer perceptron* aconsegueix trencar aquesta restricció per a classificar els conjunts de dades que no són separables linealment de forma similar al de l'esquema de la dreta a la *Figura 4.2.1*.

Els MLP tenen les mateixes capes de sortida que d'entrada, però poden tenir diverses capes ocultes (*hidden layers*²⁰) entre les capes esmentades (*Figura 4.2.2*).

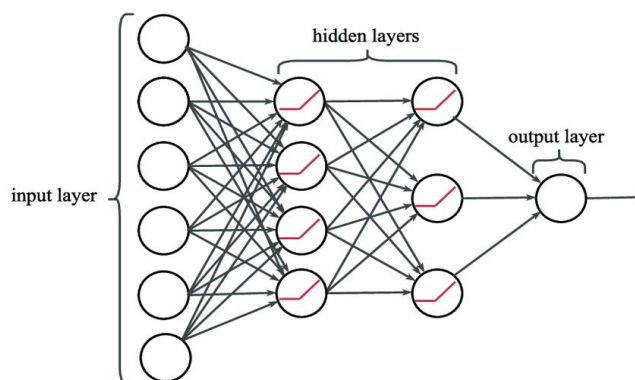


Figura 4.2.2. Esquema de xarxa neuronal

Els *MLP* empren funcions d'activació en cadascuna de les capes calculades. Una funció d'activació normalment utilitzada és la sigmoide, ja que es continua i derivable, i permet la retropropagació (Koskela, T, et al 1996). Els *MLP* constitueixen la base de les xarxes neuronals.

¹⁹ *XOR* és una porta lògica que fa referència a *EXCLUSIVE OR* en anglès realitza la funció booleana $A'B + AB'$. És una porta on si i només si una entrada és *ON*, la sortida serà activada.

²⁰ En les xarxes neuronals, un *hidden layer* està situat entre l'input i l'output de l'algoritme. El nombre d'unitats de nodes ocults controla el nombre de pesos disponibles per l'entrenament de la xarxa neuronal. Es caracteritzen per no estar connectats directament ni amb les dades entrants ni amb les de sortida.

4.3. Extreme learning machines (ELM)

Les *ELM* són una altre estructura de xarxa neuronal que es pot inicialitzar aleatòriament. L'essència rau en què els paràmetres d'aprenentatge dels nodes ocults, incloent-hi els pesos d'entrada i els biaixos, s'assignen aleatòriament, mentre que els pesos de sortida es poden determinar analíticament mitjançant la simple operació inversa generalitzada. El paràmetre que cal definir és el nombre de nodes ocults. En comparació amb altres algorismes, les *ELM* proporcionen una velocitat d'aprenentatge molt més ràpida i un millor rendiment de generalització (Figura 4.3.1). (Ding, S, et al., 2014)

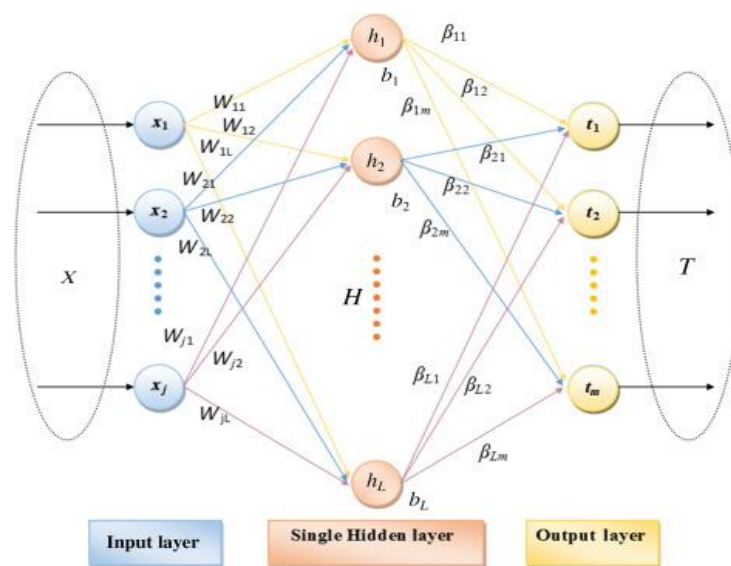


Figura 4.3.1. Esquema d'ELM

5. ELABORACIÓ PRELIMINAR DEL BOT

5.1. Definicions i funcions prèvies

El primer pas és definir tot una sèrie de funcions que serveixin per a donar indicacions al bot sobre quan ha de comprar i quan ha de vendre. Es realitzaran prediccions amb les tres metodologies definides anteriorment: *ARIMA*, *MLP* i *ELM*.

La primera limitació a l'hora de seleccionar quins models *ARIMA* poden predir bé un període determinat és l'elecció correcte dels paràmetres p i q . Tanmateix, com que no hi haurà cap individu per a cada moment t de temps observant els gràfics d'*ACF* i *PACF*, s'haurà d'escollir algun indicador que ajudi al bot a escollir quin és el millor model per a predir el preu de cotització a l'instant $t+1$.

L'única idea possible a l'abast del nivell de coneixement de programació adquirit va ser generar un vector de permutacions anomenat *valors*. Hagués estat interessant poder crear una funció que avalués comportaments gràfics de les sèries en cada moment del temps, però la complexitat hagués estat enorme. Ja existeixen algunes funcions com per exemple l'*auto.arima()*²¹ que ja tenen present aquesta informació i volia intentar provar alguna cosa que no s'hagués provat abans.

- **Vector *valors***: Generació de valor amb totes les permutacions possibles entre 0 i 5 per als valors p i q que s'utilitzaran per a construir tots els models *ARIMA*.

Mitjançant aquest vector es calculen, emprant les dades fins a $t-1$, tots els models *ARIMA* possibles amb els valors de p i q esmenats. Seguidament, es calcula la predicció per al valor de t , es calcula i s'emmagatzema la diferència en valor absolut i percentual entre el valor predit per a l'instant t i el valor real en l'instant t . Es tindrà, doncs, seleccionat el model per a calcular la predicció per al període $t+1$ lligat directament a l'operació compra-venta que s'efectuarà.

La funció que efectua el procediment descrit és la següent:

- ***ARIMAS(bitc, valors, x)***: Funció creada per avaluar tots els possibles models *ARIMA* per als valors de cotització $N-1$ fins a N , corresponents als moments de temps $t-1$ fins a t . Es seleccionen tots aquells models que tinguin un valor predit de tendència de 1 (encert) per aquest període i es tria el de menor *EP*. Pren com a arguments *bitc* (sèrie de valors de cotització de Bitcoin dels últims períodes que s'indiquin fins a t), el vector *valors* i finalment x , que serà 1 sempre, atès que només es prediu una observació en cada cas.

²¹ La funció *auto.arima()* de la llibreria *forecast* d'R proporciona una opció ràpida per construir pronòstics amb sèries temporals ja que avalua entre tots els possibles models, el millor considerant diversos criteris: estacionarietat, estacionalitat i diferències, entre altres.

Per tant, aplicant la funció *ARIMAS*, obtenim els valors de p i q que han obtingut el mínim *EP* i una predicció de tendència adient. Amb aquests valors es construirà un nou model amb les dades de I fins a N i es realitzarà la predicció per a l' instant de temps $t+1$.

D'aquesta manera es pot elaborar un *ranking* amb els models que han obtingut un millor valor en la predicció per a N .

No obstant, es podria donar el cas que un model hagués obtingut un error percentual molt baix, però que no hagués predit correctament el moviment de tendència. Per exemple: suposem que l'últim valor de la cotització del Bitcoin ha estat de 40.000€ i la del penúltim, 39.000€. Pot ocórrer que un model hagi predit per al moment t (referent al 40.000€) que el Bitcoin cotitzaria a 43.000€ i que un altre model hagi predit que cotitzaria a 38.500€. Si calculéssim els errors percentuals absoluts d'ambdós, obtindríem que, per al primer cas, hi ha un *EP* de $(3.000/40.000) = 0,075\%$. Per al segon model s'obtindria un *EP* de $(1.500/40.000) = 0,0375\%$. A priori, el segon model s'ha quedat més a prop de la predicció correcta i té un *EP* menor, no obstant el primer model ha predit una pujada i el segon una baixada, i és el primer el que ha predit bé la tendència. Per al tipus de problema al que es fa front, és prioritari predir bé la tendència abans que obtenir un *EP* baix. Això s'explica perquè, si es prediu una pujada, s'executarà l'ordre COMPRA i si es prediu una baixada s'executarà l'ordre VENDA. Per tant, sempre tindrà més pes encertar la tendència correcta que un error percentual baix. Donat aquest marc, a l'hora d'elaborar el *ranking* amb els models que han obtingut un millor valor en la predicció per a n , només es prendran en consideració aquells que hagin predit bé la tendència per al moviment de $N-1$ fins a N . Un cop filtrats, només s'emmagatzemaran en un espai de memòria temporal aquells valors de p i q del model que tinguin amb el menor *EP*.

Per a l'elaboració del segon mètode, s'utilitzen funcions d'R de la llibreria *nnfor*²².

- ***NEURAL(bitc,x)*** Funció creada per a generar una xarxa neuronal amb la funció de la llibreria *nnfor*: *mlp()*, el nombre de *hidden nodes* està fixat a (10,5).

Simplement, s'anota el valor predit per l'estimació de la funció.

Per a l'elaboració del tercer mètode, s'utilitzen de nou funcions d'R de la llibreria *nnfor*.

- ***ELM(ts(bitc))*** Funció pròpia de la llibreria *nnfor* que, conjuntament amb un *forecast*, calcula les prediccions per a aquest mètode.

²² El paquet *nnfor* és un paquet d'R que facilita la predicció de sèries temporals amb *MLP* i *ELM*. En la versió actual encara no suporta *deep learning* tot i que s'espera que en un futur no gaire llunyà ho suporti. Es basa en el paquet *neuralnet*. Aquesta llibreria és àmpliament utilitzada durant el treball, atès que dos dels models amb els que es treballa hi pertanyen.

5.2. Validació i evaluació

Per tal de poder avaluar els resultats prèviament a poder implementar el bot de *trading*, cal observar i analitzar com han estat les prediccions efectuades per cada un dels tres mètodes.

La primera estratègia per a la comparació dels tres mètodes va ser anant anotant quines eren les prediccions per a cada model. Es va efectuar de forma manual i en les hores lliures. Els resultats de les prediccions, tot i no cobrir tota la franja horària, eren seleccionats aleatòriament.

Es va obtenir un total de 27 anotacions corresponents a 27 moments del temps seleccionats aleatòriament amb els resultats presentats a la *Figura 5.2.1*:

% d'incert de tendència

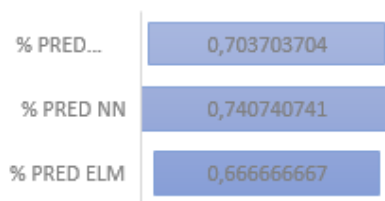


Figura 5.2.1. Percentatge d'incerts per càlcul manual

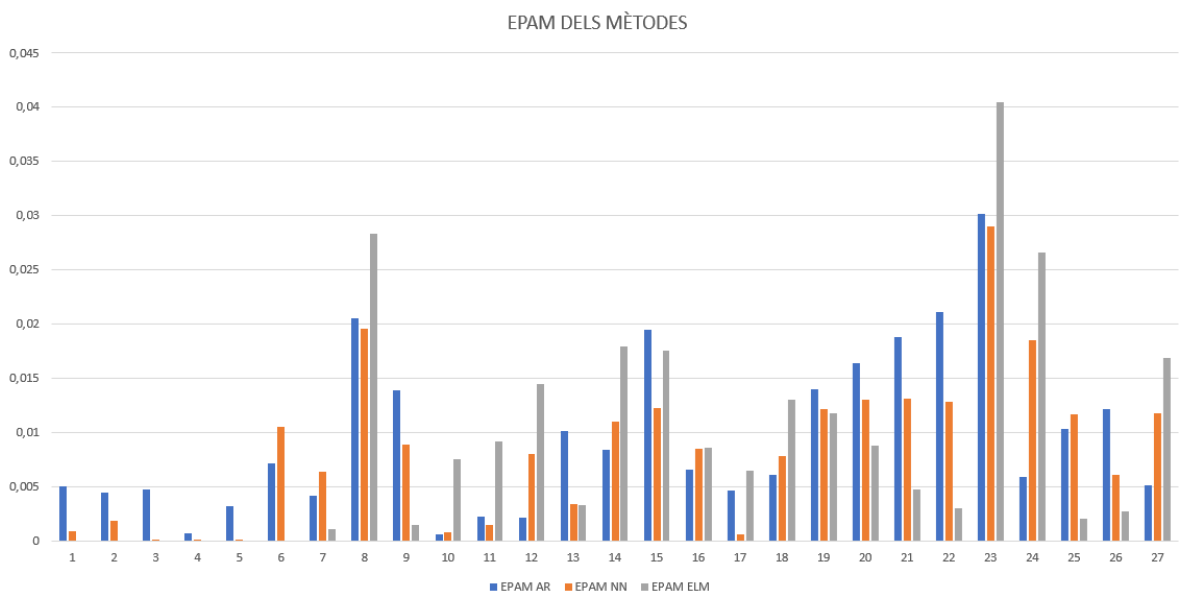


Figura 5.2.2. Gràfic dels EPAM o EP calculats manualment de cada mètode

Tant els % d'incerts de tendència com els *EP* semblen molt baixos (*Figura 5.2.2*). Aquests resultats il·luminaven un camí molt esperançador respecte a la viabilitat d'aplicació del bot. Tanmateix, es podria donar el cas que algun dels models tingués un biaix sistemàtic a l'hora de fer prediccions. Es va decidir, doncs, visualitzar el gràfic dels % errors però sense el valor absolut.

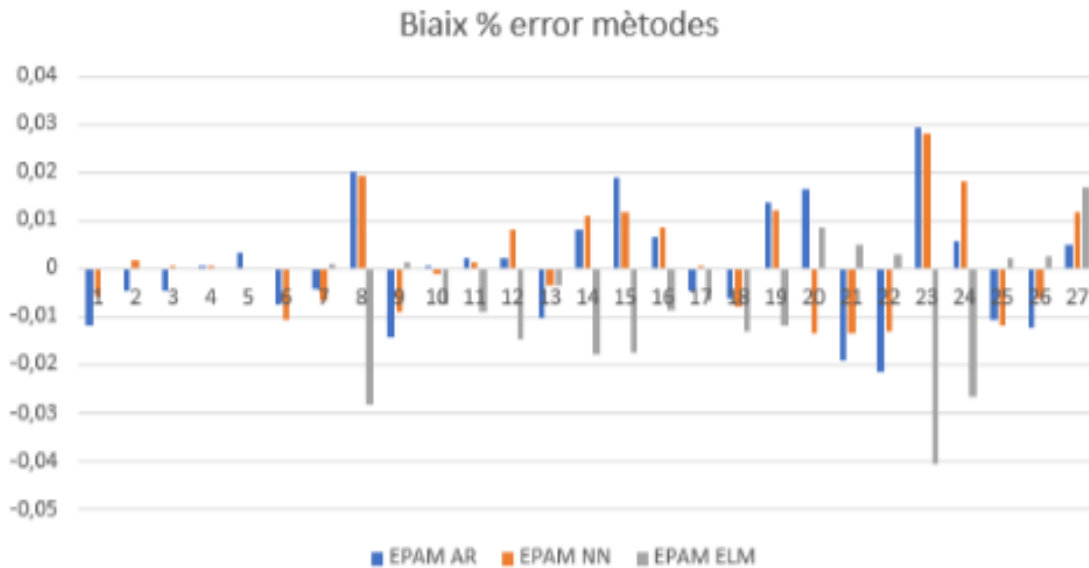


Figura 5.2.3. EPAMS o EPs en valor absolut calculats manualment

Al gràfic de la *Figura 5.2.3* s’observa clarament que hi ha un biaix a l’hora de realitzar les prediccions. Per al cas de les prediccions realitzades amb *NN* i amb *ARIMA*, sembla que no hi ha una tendència determinada. Per tant, es pot excloure que mai tendeix a sobre predir o a “infrapredir”. Aquest efecte s’observa per l’aparent balanç. En canvi, per a l’*ELM* s’observa clarament que en general tendeix a “quedar-se curt” en quant a les prediccions. És a dir, sembla que en tots els casos no s’acaba d’adequar mai bé.

No obstant, no és difícil adonar-se de que els % de predicció i els errors no acaben de ser representatius. Una mostra de només 27 casos no permet inferir-ne la resta. La idea és que el bot sigui operatiu les 24 hores del dia, tots els dies de l’any, per tant es va decidir buscar altres formes de validar-lo. La validació manual pot ser útil per tenir una lleugera idea de si els mètodes estan funcionant o no, però l’altre opció per a la que es va optar i que ha estat la definitiva del treball és la creació d’un programa de *backtesting*.

La idea és senzilla de plantejar, però complexa de dur a terme: creació d’un *software* que, mitjançant una execució, permeti avaluar a posteriori les prediccions que hagués fet el bot en un període de temps determinat. S’ha realitzat mitjançant la *GET*, definida a l’inici, que obté les cotitzacions horàries històriques, i s’ha seleccionat un període de 60 dies, per tal de tenir una idea de quina efectivitat té el bot. Un cop seleccionat aquest període, s’han de crear unes funcions que incloguin les funcions descrites per a calcular les prediccions en tots els casos per a tots els mètodes, per tal de poder avaluar del funcionament.

Per poder abordar aquesta tasca cal, però, definir una sèrie de funcions que permetin avaluar la capacitat predictiva. Les limitacions a les que s’ha de fer front són, tant l’elaboració del programa com l’alta potencia de còmput necessària, per a poder validar grans grups de dades.

- **ARIMATESTING(*bitc, bt1, n, d*)**: Funció creada per a calcular les prediccions per a tot un període $1:N$ (depenent de la crida total de la funció d'obtenció de cotitzacions mitjançant trucada *API*). El *data frame bitc* conté la informació de totes les cotitzacions, el *data frame bt1* conté tota la informació obtinguda per trucada *API*, temps, cotització, volatilitat, etc. El valor de d indica l'amplitud de sèrie que s'utilitza per a construir cada predicció. Per un valor de 150, que és el que s'ha seleccionat, totes les sèries que es construïran en cada moment del temps per a validar el bot, tindran una longitud de d . Anant des de i fins a $d+i$. L'*output* és un *csv* escrit que conté les prediccions, el model seleccionat en cada iteració, i un valor assignat que és 0 o 1, segons si s'ha encertat el moviment de tendència (pujada, baixada) del període següent.

Aquesta funció genera J prediccions. Per a realitzar cada predicció, s'empren les 150 observacions d'hores prèvies. Després d'anotar per a cada període quina és la predicció que hagués obtingut el bot, quins errors té i si ha encertat o no per a cada període, la funció escriu un CSV per a poder ser consultat un cop s'acabi d'executar tot el programa. El que s'aconsegueix és automatitzar tot el procés, que s'havia intentat de forma manual, exposat en l'apartat anterior. És una funció que requereix molta potencia de còmput, atès que per a cada iteració cal calcular tots els *ARIMA* de 0 a 5 (valors p i q) i seleccionar-ne l'òptim; a partir d'aquest òptim, calcular la predicció per al període següent, comparar el període predit amb el real, anotar resultats i, finalment, escriure un CSV. Dins de la funció, se'n crida una altre anomenada *ARIMAS2()*. És una modificació de la funció *ARIMAS()* que calcula les prediccions per al bot.

- **NNTESTING(*bitc, n, d*)**: Funció creada per a calcular les prediccions per tot un període $1:n$. L'*output* generat és un *csv* amb totes les prediccions per a cada període emprant la funció *mlp* de la llibreria *nnfor*. S'assigna un 1 o un 0 a cada fila, segons si s'ha encertat la tendència. Utilitza la funció *NEURALPRED()* definida prèviament, per a realitzar les prediccions.

La funció és exactament la mateixa que en el cas anterior, però emprant xarxes neuronals enlloc dels models *ARIMA*.

- **ELMTESTING(*bitc, n, d*)**: Funció creada per a calcular totes les prediccions per un període $1:n$. L'*output* generat és un *csv* amb totes les prediccions per a cada període emprant la funció *elm* de la llibreria *nnfor*. S'assigna un 1 o un 0 a cada fila, segons si s'ha encertat la tendència.

La funció és exactament la mateixa que en el cas anterior, però emprant *Extreme Learning Machines* enlloc dels models *ARIMA*.

En un primer moment, la intenció era que per a cada predicció N èssima, s'utilitzessin totes les observacions prèvies, és a dir, que per a predir el valor de la cotització per al període $N+1$, s'empressin les observacions de 1 fins a N ; per a predir el valor de la cotització per al període $N+2$, s'empressin totes les observacions de 1 fins a $N+1$, etc. No obstant, s'ha decidit no

realitzar-ho d'aquesta forma ja que per a cada observació predita s'haurien hagut d'utilitzar un nombre diferent d'observacions i, a més, cada predicció tindria un cost computacional major a l'anterior, divergint totalment el temps d'operació de la funció que, ara per ara, ja es demora més de 4 hores per a poder executar-se i validar un període d'uns 60 dies. D'aquesta manera, assegurem, tant la viabilitat de la validació com una aproximació real degut a la naturalesa dinàmica del programa, en què el nombre d'observacions és fix per a cada iteració amb un inici i final mòbils.

El cost computacional total per a validar uns tres mesos operatius del bot a posteriori, per tal d'entendre quina fiabilitat té, és d'aproximadament unes 7 hores amb un ordinador relativament potent. Tot i les limitacions, és molt més factible validar el model seguint aquests passos que anotant individualment en moments de temps convenients, però aleatoris.

L'elecció de la franja horària ha estat la mateixa per a les tres metodologies predictives. En total, s'ha decidit realitzar 1440 prediccions. És a dir, es disposarà de tres *csv* amb tota la informació sobre les cotitzacions i prediccions per a cada tipus de model amb els errors i encerts corresponents. El valor de *d*, referent a la longitud fixada per a realitzar cada predicció, que compondrà una línia sencera al *CSV*, ha estat fixat en 150 observacions. Un nombre suficientment gran com per a obtenir una mostra representativa i amb la que tingui sentit treballar en cada iteració, i a la vegada suficientment "petita" com a per a poder executar el *backtesting* a nivell de càlcul de computació.

6. RESULTATS I DISCUSSIÓ

6.1. Programa Backtesting

Un cop renderitzats els tres CSV podem procedir a avaluar com han estat les prediccions que hagués fet el bot per cada moment de temps. En total s'avaluen 1440 prediccions que corresponen a 60 dies. Primer es mostra un gràfic de l'evolució de la cotització horària per aquest període (*Figura 6.1.1*). El període amb el que s'ha fet el *testing* és des de 22:00:00 (6/4/2021) fins 22:00:00 (5/6/2021).

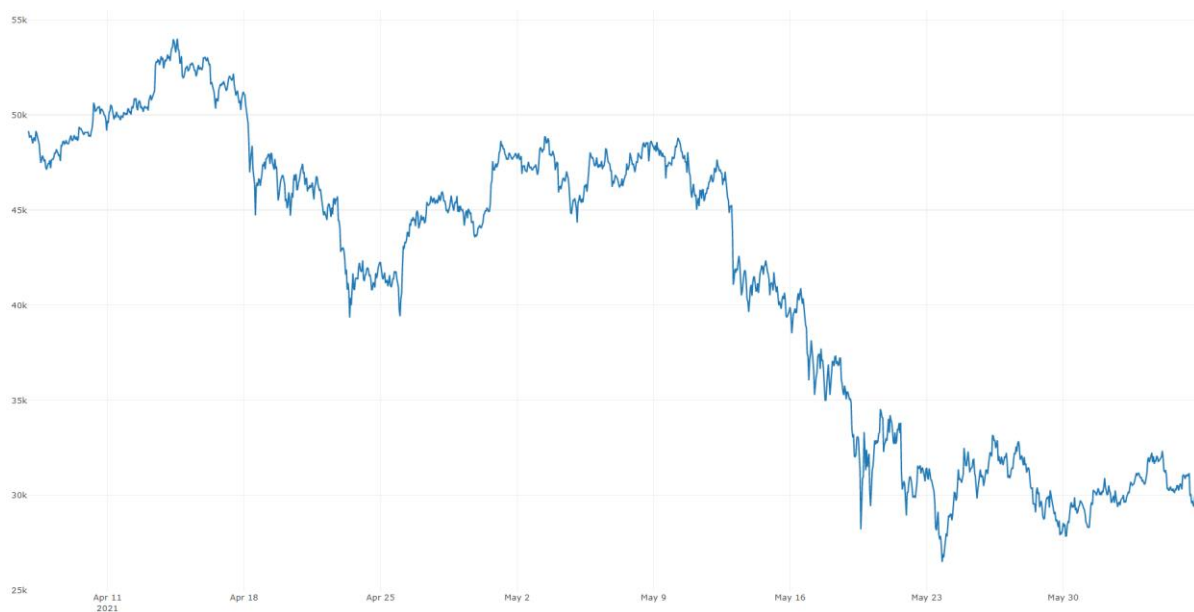


Figura 6.1.1. Gràfic de la cotització del Bitcoin per al període estudiat

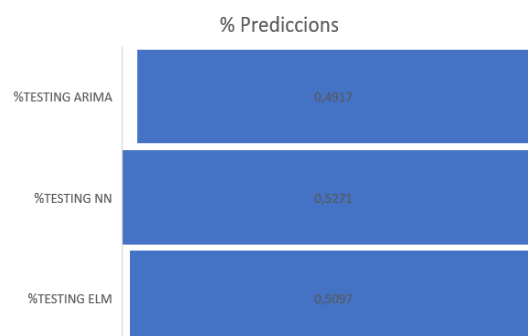


Figura 6.1.2. Percentatge de prediccions encertades calculades per simulació

S'observa que el % d'encert de tendència és molt similar (*Figura 6.1.2*). Malauradament, tots al voltant del 50%. Això indica que, a priori, el bot no seria massa útil, atès que encerta amb la

mateixa freqüència que ho faria una moneda. No obstant, si s'analitza en quines situacions s'ha encertat segons quins models, es podria arribar a elaborar una estratègia guanyadora que, en comptes d'operar en tots els moments de temps, compli només sota unes certes condicions.

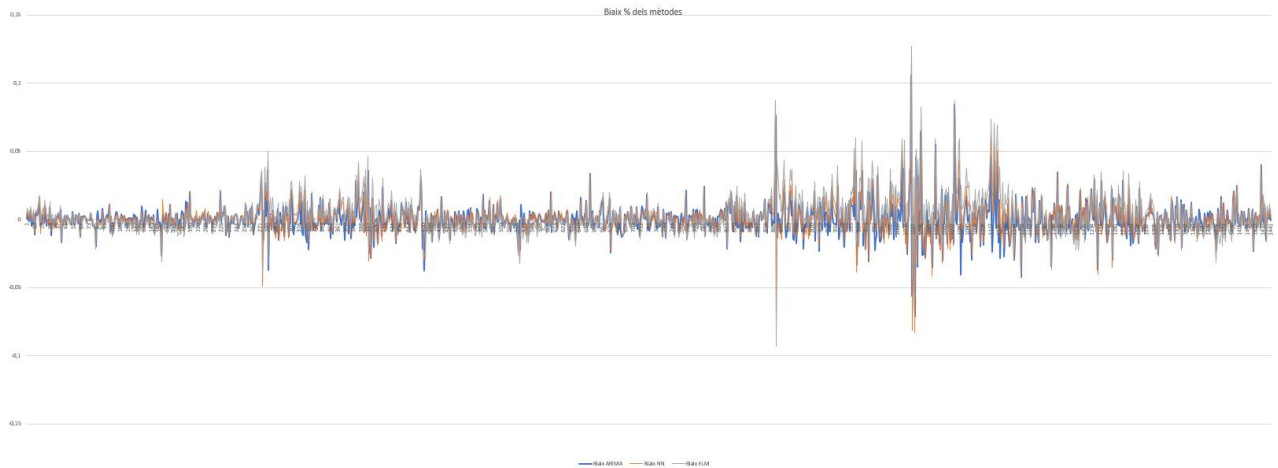


Figura 6.1.3. Gràfic dels errors percentuals de cada mètode calculats per simulació

A la Figura 6.1.3, la llegibilitat es complica degut al alt nombre d'observacions, una forma per a poder visualitzar com han realitzat les prediccions els tres mètodes i entendre si hi ha o no presència de biaixos sistemàtics és graficar els histogrames dels errors percentuals per cada un dels mètodes (Figura 6.1.4). Es pot observar gràficament que no sembla que hi hagi gaires diferències entre els errors dels mètodes.

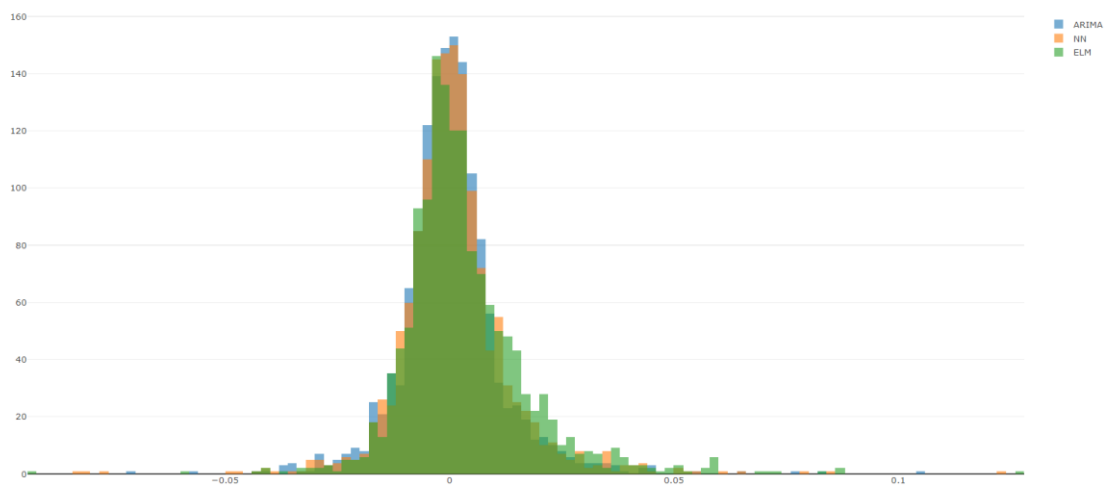


Figura 6.1.4. Distribució dels errors dels tres mètodes.

Es pot elaborar una taula descriptiva comparativa per als tres mètodes (Taula 6.1.1).

Taula 6.1.1. Descriptiva de la serie de rendibilitats per a cada mètode

ARIMA		NN		ELM	
nbr.val	1441.00000	nbr.val	1441.00000	nbr.val	1441.00000
nbr.null	0.00000	nbr.null	0.00000	nbr.null	0.00000
nbr.na	0.00000	nbr.na	0.00000	nbr.na	0.00000
min	-0.07114	min	-0.08309	min	-0.09281
max	0.10589	max	0.12215	max	0.12678
range	0.17703	range	0.20524	range	0.21959
sum	0.58381	sum	1.36661	sum	4.76748
median	-0.00005	median	0.00008	median	0.00092
mean	0.00041	mean	0.00095	mean	0.00331
SE.mean	0.00031	SE.mean	0.00034	SE.mean	0.00038
CI.mean.0.95	0.00061	CI.mean.0.95	0.00066	CI.mean.0.95	0.00074
var	0.00014	var	0.00016	var	0.00021
std.dev	0.01176	std.dev	0.01281	std.dev	0.01438
coef.var	29.01760	coef.var	13.51226	coef.var	4.34547

A la Taula 6.1.1, es pot observar una descriptiva completa sobre els biaixos per cada un dels mètodes. En primer lloc, crida l'atenció que totes les mitjanes són positives. Sembla que el model *ARIMA* es el que aconseguix prediccions més precises, ja que compta amb la menor mitjana dels tres. Per contra, el model *ELM* té una mitjana molt superior als altres dos mètodes. En quant a la mediana, s'observa que pels tres casos, l'error percentual és molt proper a 0, per tant, no podem afirmar que hi hagi grans diferències entre els mètodes en quant a biaixos. En general, s'observa un comportament de biaixos que, en concordança amb el gràfic de les distribucions, deixa entreveure que no hi ha grans biaixos sistemàtics per cap dels 3 mètodes. Només hi ha dos estadístics que són dignes de menció:

Un és el coeficient de variació, que per al cas del mètode *ARIMA* és molt superior als altres dos, això s'explica més per la mitjana que per la desviació típica, ja que aquesta és en proporció, bastant més diferent que la desviació típica respecte els altres.

L'altre estadístic a considerar és la suma. Per al *ELM*, la suma de tots els biaixos té un valor molt superior als altres dos mètodes. Podem afirmar que l'*ELM* tendeix a sobre predir i, per tant, acostuma a visualitzar escenaris més optimistes que la realitat, és dels tres el cas en el que es pot afirmar que hi ha un petit biaix sistemàtic. No obstant, els tres mètodes tenen una suma i una mitjana positives, per tant, en global han sigut més aviat optimistes.

Podria ser que hi haguessin certs moments de temps en els que fos molt difícil predir correctament la tendència per al següent període. Per tal d'avaluar si existeix aquesta opció, s'ha pensat que es podria calcular en quins casos han coincidit els models per veure si hi ha una concordança entre ells i existeixen moments de temps en els que tots fallen i tots encerten. Si es vol analitzar de forma col·lectiva, es pot simplement representar quin % dels cops han encertat els 3 models, i quin % dels cops han encertat dos d'ells, un d'ells i cap d'ells (*Figura 6.1.5*).

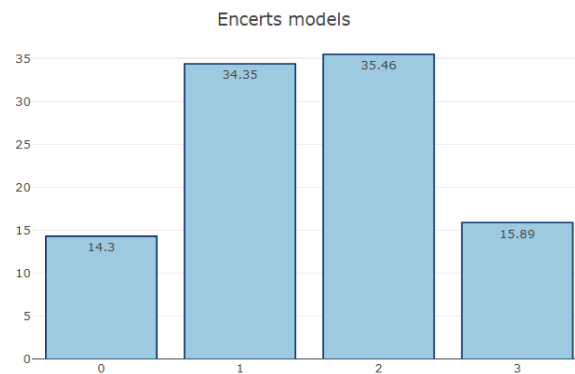


Figura 6.1.5. Gràfic de percentatge d'encerts i nombre de models

Es pot observar que el nombre de cap encert ha estat 14,3%. Tenint en compte que hi ha 3 models que poden encertar, si hi hagués una aleatorietat total, i cada model tingués un 50% de probabilitat d'encertar, la probabilitat de que els 3 models encertessin seria $0,5^3 = 0,125$. En aquest cas s'observa que tres models encerten en un 15,9% dels casos, per tant és un escenari positiu.

No obstant, caldria separar aquells casos amb increment positiu d'aquells amb increment negatiu per veure si hi ha diferències (*Figura 6.1.6*). Els casos on l'intercanvi de valor des de el moment t fins al moment $t+1$ ha estat negatiu ha sigut un 50,03% dels casos, mentre que els casos per als que ha hagut un increment positiu ha estat del 49,97%.

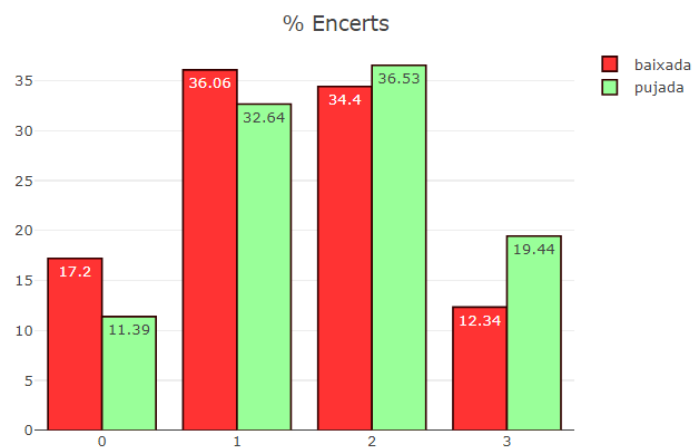


Figura 6.1.6. Gràfic de percentatge d'encerts i nombre de models estratificat per tendència

Un cop separat, a la *Figura 6.1.6* s’observa que la diferència no es distribueix de la mateixa manera. Les conclusions en quant a quan encerten tots i quan no encerta cap són completament diferents si separem o no separem per tipus de període. En períodes de pujada el 19,44% dels cops, els 3 models encerten. Quan el mercat baixa, només el 12,34% dels cops els tres models encerten la baixada.

En quant al cas 0 (cap model encerta) passa el contrari que quan els tres models encerten. En intervals de baixada, el 17,2% dels cops, cap model encerta, mentre que quan hi ha pujada, només en un 11,39% dels casos els 3 models fallen.

Aquestes anàlisis donen una imatge general sobre com es poden relacionar els tres models. No obstant, no s’ha analitzat com interactua cada un amb la resta. Una possible alternativa per valorar aquest efecte és usar un índex que ens indiqui el *matching* que hi ha entre cada parella de models. Una possible opció és emprar l’índex de *Kappa* (Ullibbarri L, et al.,1999) el qual representa la proporció d’acords observats que no pot ser explicada per l’atzar *Taula 6.1.2*.

Taula 6.1.2. Index de Kappa per a cada cas

Kappa Global	Kappa Arima vs NN	Arima vs ELM	NN vs ELM
Fleiss' Kappa for m Raters Subjects = 1441 Raters = 3 Kappa = 0.0688 z = 4.52 p-value = 6.07e-06 Kappa z p.value 0 0.069 4.524 0.000 1 0.069 4.524 0.000	Fleiss' Kappa for m Raters Subjects = 1441 Raters = 2 Kappa = 0.0434 z = 1.65 p-value = 0.0998 Kappa z p.value 0 0.043 1.646 0.100 1 0.043 1.646 0.100	Fleiss' Kappa for m Raters Subjects = 1441 Raters = 2 Kappa = 0.00624 z = 0.237 p-value = 0.813 Kappa z p.value 0 0.006 0.237 0.813 1 0.006 0.237 0.813	Fleiss' Kappa for m Raters Subjects = 1441 Raters = 2 Kappa = 0.156 z = 5.93 p-value = 2.94e-09 Kappa z p.value 0 0.156 5.935 0.000 1 0.156 5.935 0.000

Taula 6.1.3. Interpretació d’índex de Kappa

kappa	grado de acuerdo
< 0,00	sin acuerdo
0,00 - 0,20	insignificante
0,21 - 0,40	mediano
0,41 - 0,60	moderado
0,61 - 0,80	sustancial
0,81 - 1,00	casi perfecto

Atenent-nos a la *Taula 6.1.3* amb els índexs *Kappa* calculats, i contrastant amb les interpretacions que te el valor de l’índex expressat a la *Taula 6.1.3*, s’arriba a la conclusió que en cap dels casos hi ha concordança entre els models. Per tant, es demostra analíticament que de forma global, els models prediuen generalment de formes diferents. Es pot afirmar doncs que en cap cas un model és substituïble per un altre de forma global, per tant, pot ser que alguns siguin millor que altres, de forma permanent o situacional.

6.2. Valor acumulat en Cartera

Sota suposició d'operació continua, es mostrarà el rendiment acumulat de la cartera si s'haguessin realitzat totes les operacions. Per tal de poder calcular-ho, es genera un programa que simuli totes les operacions.

6.2.1. Percentatge d'encerts

En primer lloc s'analitzarà breument quin tipus d'operació ha recomanat cada model en cada cas. En cas que la predicció per al valor $t+1$ de cada model fos d'un valor superior al N , la recomanació d'aquest hauria estat de COMPRA, altre cas de VENTA. A la *Taula 6.2.1.1* es mostren els % per cada tipus d'operació recomanada on C indica COMPRA, i la V indica VENDA.

Taula 6.2.1.1. Taula amb percentatge de compres i vendes predites per cada model

ARIMA			NN			ELM		
	Frequency	Percent		Frequency	Percent		Frequency	Percent
C	710	49.27	C	777	53.92	C	834	57.88
V	731	50.73	V	664	46.08	V	607	42.12
Total	1441	100.00	Total	1441	100.00	Total	1441	100.00

Les opcions recomanades varien lleugerament entre models. Mentre que per al *ARIMA* en el 50% dels casos recomana comprar i en l'altre 50% vendre, per als models *NN* i *ELM* tendeix a recomanar més sovint comprar que vendre.

Anteriorment s'ha vist el % d'encert per a cada mètode i els seus biaixos. El factor més important de tots per un bot de *trading* no és tant el % d'encert, sinó l'esperança matemàtica del total d'operacions. Per exemple, si encerta només el 45% de les vegades però cada encert suposa el doble de benefici que cada no encert, el bot tindria una esperança positiva ($E[X] > 0$): $0,45 \cdot 2x - 0,55x = 0,35x$.

La forma correcta i definitiva per a conèixer quin ha estat el rendiment del bot passa per calcular el valor de cartera que s'hagués obtingut per a cada instant de temps si aquest ja portés un temps funcionant. Amb els *CSV* generats per les funcions exposades també es tenen les dades de les cotitzacions de cada moment a part de les prediccions. S'ha creat un programa a posteriori que calcula totes les plusvàlues i pèrdues que s'haguessin obtingut si s'hagués obeït el patró de decisió escollit per a cada un dels tres mètodes. El càlcul es realitza d'una forma senzilla. Suposant que en el moment de temps t un dels mètodes calculi que la predicció és un valor superior a l'actual, emetrà un missatge de COMPRA. En tal cas, emmagatzemarem la informació del valor de la cotització en el moment t . Calcularem la diferència de preu entre el valor de cotització en el moment $t+1$ el valor de cotització en el moment t . Aquesta diferència

serà el guany en valor absolut associat a aquella operació. El guany el relativitzarem dividint entre el valor de cotització en t per a tenir-lo en valor percentual. Bàsicament, es crearà un programa amb dues condicions: una per quan hi hagi una COMPRA i l'altre per quan hi hagi una VENDA, per tal de calcular els guanys percentuals en cada operació. Crearem un vector, anomenat S , que anirem concatenant amb si mateix, multiplicant pel $S-1$ en cada iteració. Seguint aquesta metodologia, obtindrem la sèrie de valors que indiquen la variació percentual en el valor de la cartera per cada moment de temps, de forma acumulada. Si apliquem aquest petit programa a tots els mètodes, obtindrem tres sèries amb el valor exacte que s'hagués acumulat si s'haguessin realitzat les 1440 operacions, corresponents a totes les hores que componen aquest període, si el bot ja hagués estat operant.

6.2.2. Valor portafoli interès compost

Per tal de facilitar la lectura, en comptes de mostrar el gràfic amb el valor respecte 1 guanyat, es suposarà que es disposava d'un capital inicial de 1000€ i que s'ha anat operant sobre aquest. Per tant el valor total de la cartera suposa reinversió absoluta (*Figura 6.2.2.1*).

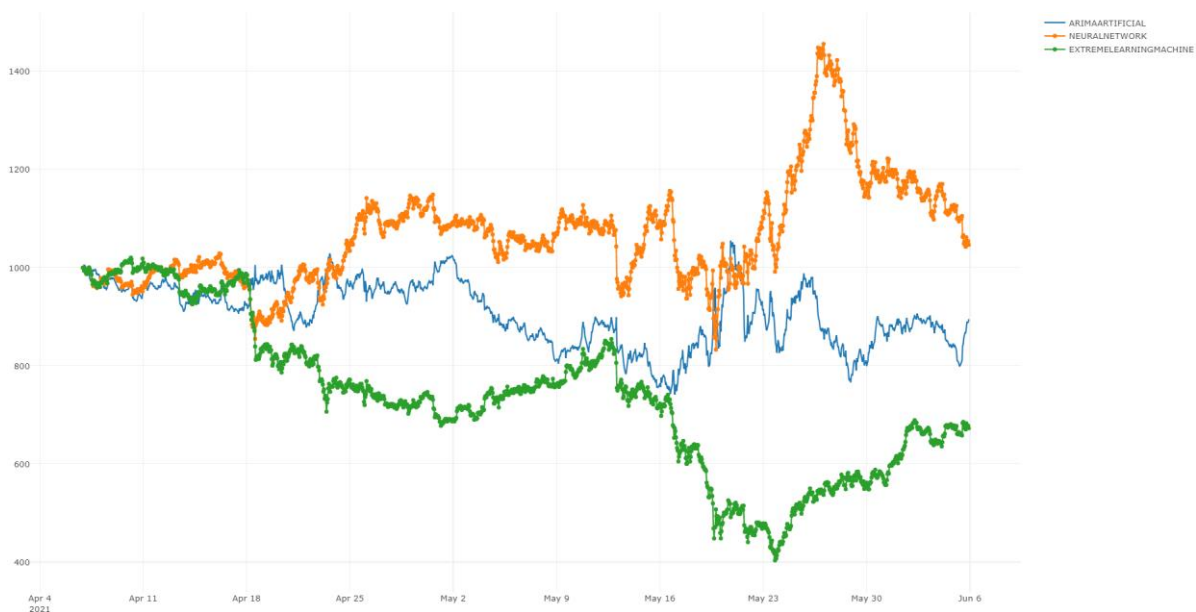


Figura 6.2.2.1. Gràfic del valor acumulat en cartera seguint cada mètode durant els últims 2 mesos

En el gràfic de la *Figura 6.2.2.1* es deixa entreveure que clarament el valor acumulat en cartera seguint les directrius indicades per el mètode d'*Extreme Learning Machines* porta a una pèrdua considerable. En concret un valor de 657€ finals respecte un valor en cartera inicial de 1000€. Per tant comporta una pèrdua de 34,3%. El mètode *ARIMA*, tampoc és massa bo. En pocs moments del temps ha provocat una cartera positiva i ha finalitzat el període amb un valor total de 895€, suposa una pèrdua de 105€ i un 10,5%. Finalment la cartera obtinguda mitjançant les directrius del mètode de xarxes neuronals (*NN*) ha obtingut un valor de 1046€, que suposa un guany de 46€ i un 4,6% de benefici en 2 mesos.

Taula 6.2.2.1. Taula descriptiva de la sèrie del valor en cartera acumulat per cada mètode

ARIMA		NN		ELM	
nbr.val	1441.00	nbr.val	1441.00	nbr.val	1441.00
nbr.null	0.00	nbr.null	0.00	nbr.null	0.00
nbr.na	0.00	nbr.na	0.00	nbr.na	0.00
min	741.53	min	833.00	min	403.39
max	1055.11	max	1456.21	max	1020.55
range	313.59	range	623.21	range	617.16
sum	1306797.09	sum	1536551.32	sum	1061539.43
median	911.15	median	1059.15	median	738.06
mean	906.87	mean	1066.31	mean	736.67
SE.mean	1.67	SE.mean	2.76	SE.mean	4.06
CI.mean.0.95	3.28	CI.mean.0.95	5.41	CI.mean.0.95	7.97
var	4038.04	var	10974.60	var	23783.79
std.dev	63.55	std.dev	104.76	std.dev	154.22
coef.var	0.07	coef.var	0.10	coef.var	0.21

Si observem els estadístics calculats per les tres sèries de la *Taula 6.2.2.1*, es pot extreure informació interessant. Per al mètode *ARIMA* observem que el valor mig en cartera des de l'inici ha estat de 906,8€. Només ha tingut un període curt amb guanys i un valor màxim acumulat de 1055€. El valor mínim ha estat de 741€ i el rang de 313€. El mètode *ELM* ha obtingut encara pitjors resultats. Un valor en cartera mitjà de 736,7€. Un valor considerablement inferior a l'inicial. Només ha tingut un màxim de 1020, però que no ha estat capaç de mantenir. El valor mínim al que ha arribat és baixíssim, 403€. Creuar un portafoli un 60% per molt que sigui operant de forma automàtica dona vertigen. Finalment, el mètode guanyador ha estat el de Xarxes Neuronals. Ha obtingut una més que satisfactòria mitjana de 1066€. És la única que està per sobre del capital inicial. El màxim al que ha arribat acumular ha estat de 1456€ i el mínim de 833€. És aquest doncs, el mètode amb les millors prediccions per aquest període, i el que probablement escolliríem com a prioritari en la majoria de situacions per tal de seleccionar si comprar o vendre en cada moment de temps.

6.3. Interacció entre els mètodes

6.3.1. Correlació en valor de cartera acumulat

Cal esmentar, que tot i ser un període de temps relativament curt (2 mesos) s'han realitzat moltes prediccions, concretament 1440, i a més, el mercat ha mostrat símptomes de molt alta volatilitat. Abans de finalitzar, s'ha pensat que es podrien realitzar un seguit de comprovacions per a descartar ja totalment la dependència entre models. Per exemple, una forma de compararlos és analitzar si existeix correlació entre els valors acumulats en cartera per a cada instant t

de temps. Una gran opció per verificar-ho és mitjançant un contrast d'hipòtesis de correlació. És la forma més rigorosa de fer-ho i no requereix compliment d'hipòtesis com la normalitat de les dades, homoscedasticitat, etc., atès que és un mètode no paramètric que es realitza emprant proves de permutació. L'estimació dels coeficients de correlació es realitzarà mitjançant *bootstrap* (Lunneborg, C. E., (1985))

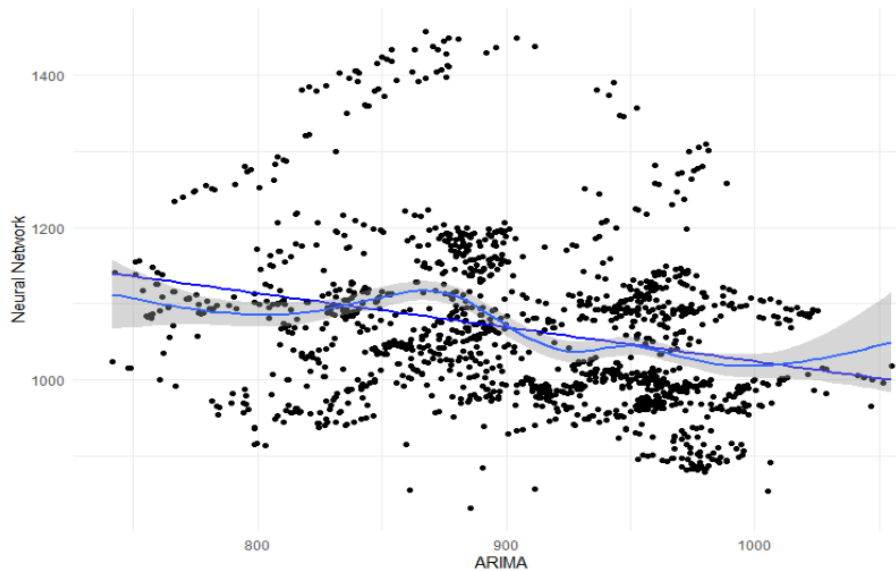


Figura 6.3.1. Gràfic de correlació entre del valor acumulat en cartera entre ARIMA i NN

A la *Figura 6.3.1* sembla que potser si que hi ha una lleugera correlació negativa entre el valor acumulat en cada moment de temps t en cartera per el mètode de xarxes neuronals i l'*ARIMA*.

Per conèixer si existeix correlació significativa, primer calculem el *p.valor* amb una prova de permutacions, fixant el nombre de permutacions en 19999. El *p.valor* obtingut ha estat molt inferior a 0,05. Concretament 0,0005. El valor estimat per al coeficient de correlació ha estat de -0,26992. Podem afirmar doncs que existeix un coeficient de correlació negatiu entre *ARIMA* i xarxes neuronals.

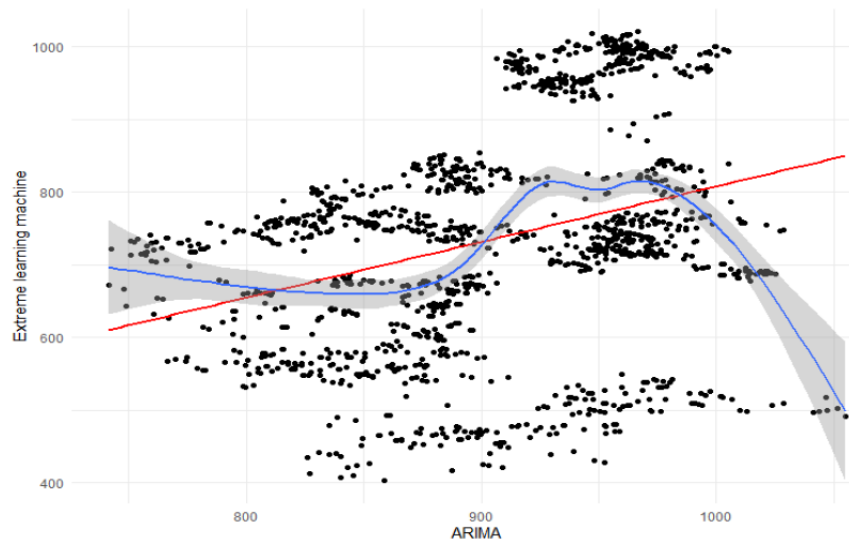


Figura 6.3.2. Gràfic de correlació entre del valor acumulat en cartera entre ARIMA i ELM

En aquest cas, també pot arribar a donar la sensació de que existeix algun grau de correlació positiva entre el valor acumulat en cada moment de temps t en cartera per el mètode ARIMA i l'ELM (Figura 6.3.2).

El p .valor obtingut ha estat molt inferior al nivell de significació 0,05. El valor estimat mitjançant *bootstrap* per al coeficient de correlació ha estat de 0,3151.

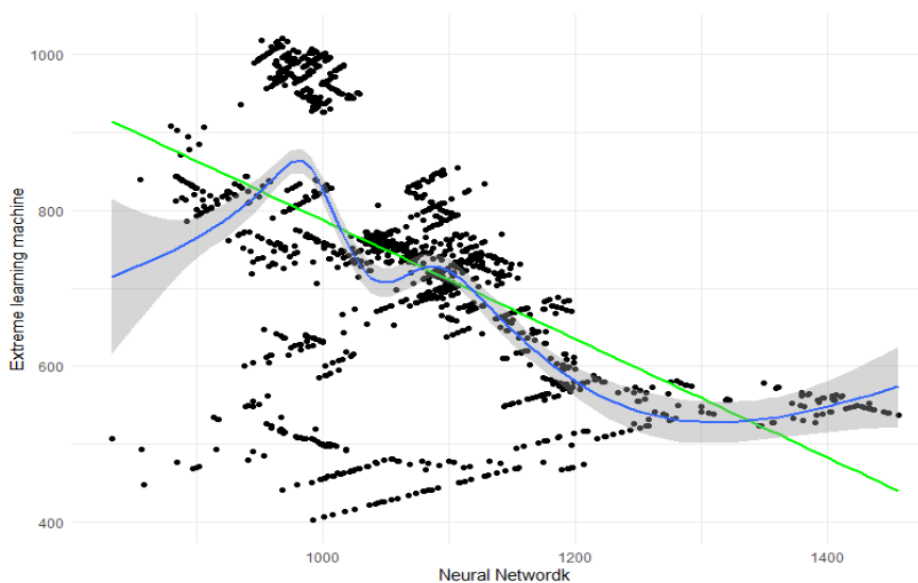


Figura 6.3.3. Gràfic de correlació entre del valor acumulat en cartera entre NN i ELM

Finalment, a la Figura 6.3.3 es mostra el gràfic d'Extreme Learning Machines vs Neural Networks. En aquest últim cas, sembla que existeix cert grau de correlació negativa. La pendent de la recta estimada és agosarada. El p .valor obtingut per a l'estimació ha estat molt inferior al nivell de significació. L'estimació *bootstrap* per al coeficient de correlació ha estat de -

0,51698. Es pot observar, doncs, que hi ha certs graus de correlació entre el valor de la cartera acumulat, segons els diferents mètodes. La utilitat de conèixer aquesta informació pot permetre en un futur si s'optimitza el bot, seleccionar combinacions de mètodes segons la situació. Si, per exemple, dos mètodes tenen una correlació positiva elevada, mai tindria sentit seleccionar el que doni una rendibilitat estimada inferior atès que serà objectivament pitjor que l'altre. No obstant, en segons quins moments, pot interessar realitzar combinacions lineals de mètodes amb correlacions negatives, per a calcular prediccions minimitzant el risc.

6.3.2. Estandardització

S'ha analitzat mètode a mètode quines eren les característiques de cadascun. Per completar aquest anàlisi podria ser interessant conèixer com interactua el valor acumulat de la cartera seguint cada un dels mètodes amb la gràfica de cotització del Bitcoin durant el període que el bot ha operat, és a dir, des de el 6 de juny fins al 5 de maig. Un mètode visual molt útil per obtenir aquesta informació és mitjançant funcions del paquet *tsviz*²³. Amb la funció *tsviz()* del paquet, es pot crear una aplicació en *shiny*²⁴ interactiva que permet estandarditzar i graficar d'una forma extremadament simple les sèries que nosaltres tinguem definides a la memòria de l'R.

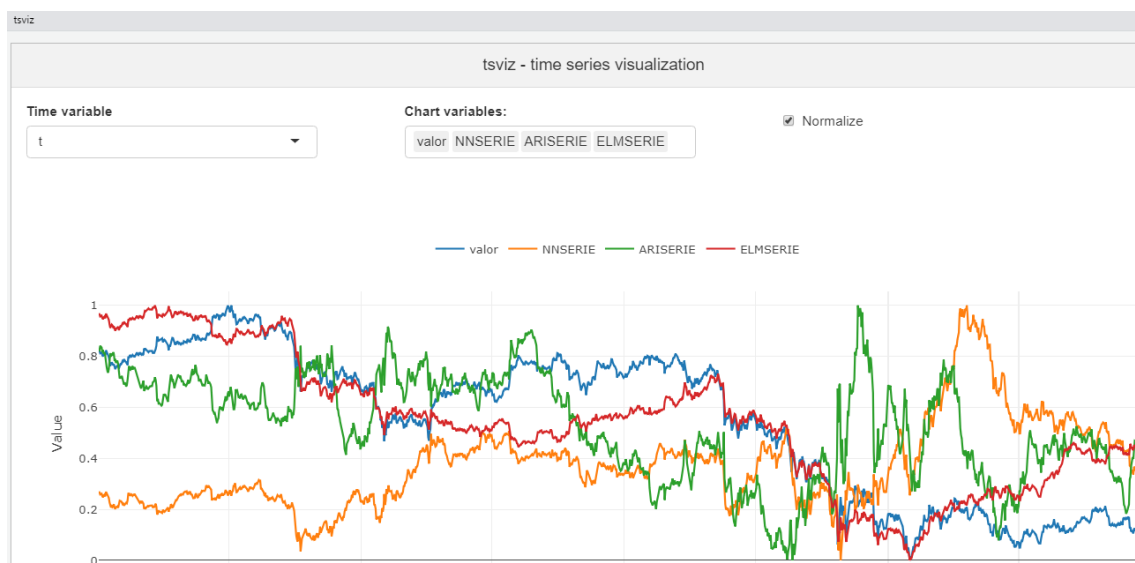


Figura 6.3.2.1. Sèries de valor en cartera acumulat per cada mètode i cotització del Bitcoin estandarditzades

A la *Figura 6.3.2.1* es pot comparar com han interactuat els valors de les carteres obtingudes seguint cada una de les metodologies, respecte a la cotització del Bitcoin en aquest mateix

²³ El paquet *tsviz* és un paquet d'R creat per visualitzar sèries temporals. Amb les funcions pròpies que conté, genera una *interface* interactiva i molt intuïtiva mitjançant la qual es poden mostrar diverses sèries temporals simultàniament, estandarditzar-les, mostrar els gràfics *ACF* i *PACF* etc.

²⁴ *Shiny* és una llibreria d'R per la construcció de quadres de mando web interactius. Permet per exemple crear *interfaces* per algoritmes o accedir i manipular taules de dades a través de controls d'*HTML*, *sliders*, botons etc. .

període. El que crida més l'atenció és que els màxim valor en cartera tant per *ARIMA* com per xarxes neuronals s'obté quan el mercat està sota mínims. És a dir, sembla que quan el mercat baixa molt durant el mes de maig, és quan millor operen aquestes dues carteres. El màxim per al mètode *ELM* es troba just al principi, i a partir de llavors, ja no torna a creuar-lo.

A continuació, es mostren els gràfics corresponents a la correlació entre els valors en cartera obtinguts per cada un dels mètodes vs el valor de la cotització del Bitcoin (*Figura 6.3.2.2*). La variable valor representa la cotització.

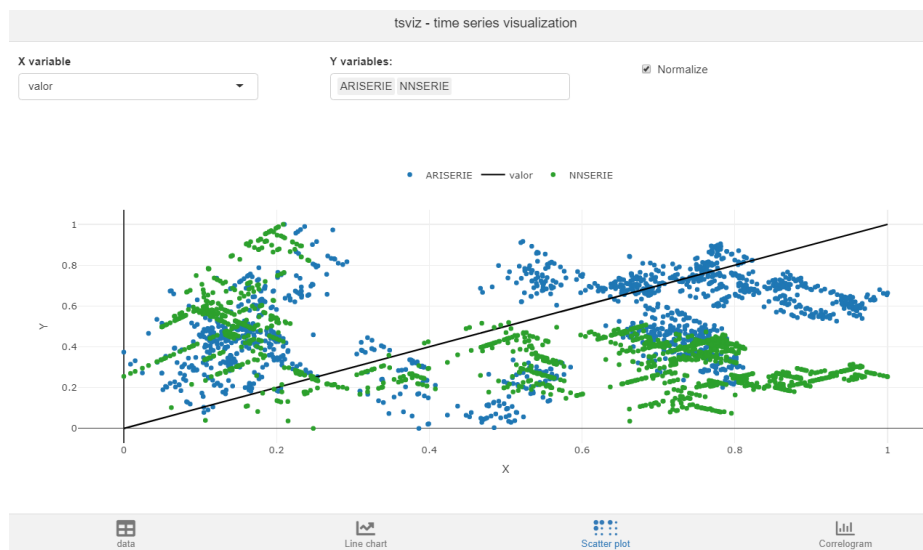


Figura 6.3.2.2. Gràfic estandarditzat d'interacció de valor en cartera seguint ARIMA i NN ajustat a la cotització

En aquest gràfic estandarditzat s'observa com correlaciona el valor de la Cartera seguint mètode *ARIMA* i Xarxes Neuronals (*ARISERIE* i *NNSERIE*) respectivament, en front de la cotització del Bitcoin. La lectura és complicada ja que no sembla que acabi d'ajustar-se gaire bé a la recta. Tanmateix, clarament, hi ha una doble modalitat. Per valors de cotització alts, el saldo de la cartera *NNSERIE* es manté en valors baixos, mentre que el de la cartera *ARIMA* es manté alts. Podem intuir doncs que el valor de la cartera seguint el mètode *ARIMA* quan la cotització és alta és superior a quan és baixa. Per altre banda, el valor de la cartera seguint el mètode de Xarxes neuronals és alt quan la cotització és baixa i baix quan la cotització és alta.



Figura 6.3.2.3. Gràfic estandarditzat d'interacció de valor en cartera seguint ARIMA i ELM ajustat a la cotització

Per a poder interpretar correctament la *Figura 6.3.2.3*, s'han hagut de separar els resultats d'*ELM*, ja que sinó es solapaven molts punts i la lectura es feia complicada. En quant al saldo obtingut seguint el mètode *ELM*, observem que el valor en cartera s'ajusta força al valor de la cotització. Per tant, és una cartera que en principi no difereix gaire de mantenir la posició estàtica des de el primer dia fins l'últim.

És interessant a part de les eines gràfiques mostrar l'estimació per al coeficient de correlació entre el valor acumulat en cartera de cada mètode versus la cotització del Bitcoin per cada un dels mètodes. De nou, es realitzarà l'estimació emprant mètodes no paramètrics (estimació bootstrap) atès que les dades no segueixen una distribució normal.

Taula 6.3.2.1. Taula amb estimacions de coeficient de correlació entre cotització i valor acumulat en cartera per cada mètode i el pvalor associat.

	ARIMA	NN	ELM
Pvalor (95%)	0	0	0
Estimació correlació	0,3875	-0,5254	0,86

Els resultats obtinguts a la *Taula 6.3.2.1* van en concordança amb les conclusions prèvies extretes amb les eines gràfiques. La utilitat de conèixer la interacció entre la cotització del Bitcoin i el valor acumulat en cartera és, principalment, conèixer quina seria la millor metodologia per aplicar, segons la previsió de mercat alcista o baixista en un plaç determinat. No obstant, no s'ha d'oblidar que, tot i tenir una correlació negativa, el valor acumulat en cartera seguint el mètode de *NN* amb la cotització del Bitcoin, segurament, els rendiments són superiors que amb els altres mètodes. Anteriorment, s'ha observat que, en valor absolut, el valor en cartera seguint el mètode de xarxes neuronals era, en gairebé qualsevol instant de temps, superior al valor acumulat per qualsevol d'altres mètodes. Tanmateix, si que pot ser útil per a decidir si no operar en cas de que les previsions de mercat per a la cotització siguin molt positives.

6.4. Anàlisi rendibilitats simples

Una altre alternativa per al funcionament del bot seria realitzar les operacions sota el no supòsit d'interès compost. És a dir, no reinvertir l'import exacte del valor de la cartera en cada operació sinó, per exemple, invertir sempre la mateixa quantitat en cada una de les compres i vendes però sense influir en les operacions futures. Suposarem el cas en el que s'hagin invertit 1000€ en cada una de les operacions. Seria difícil gestionar tantes operacions amb un import fix, i a priori, si el bot és bó i consistent, no tindria cap sentit operar a interès fix.

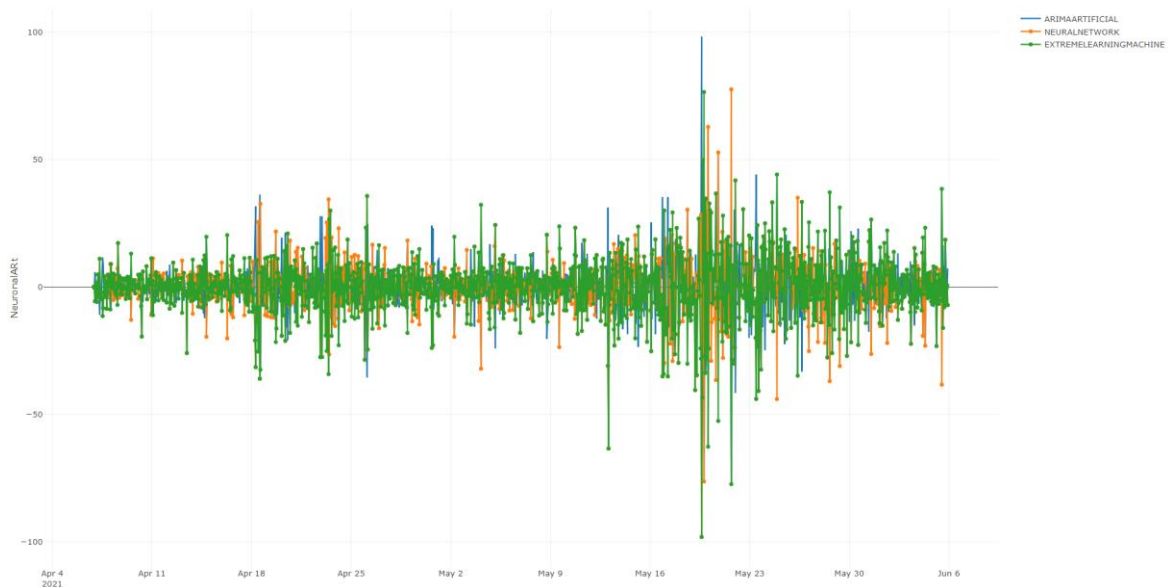


Figura 6.4.1. Gràfic de rendibilitats simples obtingudes en cada operació

A la *Figura 6.4.1* es mostren les rendibilitats simples obtingudes per a cada operació. És un gràfic de difícil lectura, però deixa intuir que, amb l'*Extreme Learning Machine*, la rendibilitat simple obtinguda en cada període és més volàtil. Per tal de conèixer més a fons com han estat, es poden descriure amb una taula. El patró que s'observa, comú per als tres mètodes, és que en el període en el que hi va haver una caiguda pronunciada en el valor de cotització del Bitcoin, les rendibilitats simples també han obtingut una oscil·lació superior, trencant amb la homoscedasticitat que semblava que la caracteritzava en els períodes previs.

Amb la *Taula 6.4.1*, es poden tenir unes nocions sobre com han estat les rendibilitats simples obtingudes per cada un dels mètodes. Com era d'esperar, el cas de *NN* és el que ha obtingut millors resultats seguit per l'*ARIMA* i el *ELM*. El que sobta és que el valor de cartera acumulat a final de període total mostra un rendiment superior a quan operàvem amb interès compost. Aquest valor es pot obtenir a partir de la suma (atès que les operacions són independents entre elles i d'import fix). La desviació típica i el rang són pràcticament iguals per a tots els mètodes.

Finalment, també sorprèn molt que la mediana del *ELM* sigui positiva, això implica que en més la meitat d'operacions amb el mètode *ELM* s'han guanyat diners.

Taula 6.4.1. Descripció de la sèrie de rendibilitats simples obtingudes per cada mètode

ARIMA		NN		ELM	
nbr.val	1441.0000	nbr.val	1441.0000	nbr.val	1441.0000
nbr.null	1.0000	nbr.null	1.0000	nbr.null	1.0000
nbr.na	0.0000	nbr.na	0.0000	nbr.na	0.0000
min	-77.4499	min	-98.1682	min	-98.1682
max	98.1682	max	77.4499	max	76.4015
range	175.6181	range	175.6181	range	174.5698
sum	-15.9074	sum	140.9765	sum	-300.5755
median	-0.1551	median	0.4635	median	0.1390
mean	-0.0110	mean	0.0978	mean	-0.2086
SE.mean	0.3024	SE.mean	0.3024	SE.mean	0.3024
CI.mean.0.95	0.5932	CI.mean.0.95	0.5932	CI.mean.0.95	0.5931
var	131.7893	var	131.7799	var	131.7459
std.dev	11.4800	std.dev	11.4795	std.dev	11.4781
coef.var	-1039.9326	coef.var	117.3388	coef.var	-55.0274

6.4.1. Comparació rendibilitats simples amb compostes

Es poden comparar els resultats de valor en cartera sota els dos supòsits, primer la cartera inflada amb 1000€ inicials i suposant reinversió absoluta, i en segon lloc la cartera resultant de realitzar totes les operacions amb 1000€ (Taula 6.4.1.1).

Taula 6.4.1.1. Comparativa entre valor en cartera obtingut per cada mètode seguint estratègia d'interès simple versus compost

INTERÈS	ARIMA	NN	ELM
SIMPLE	984,09	1140,977	699,42
COMPOST	895,14	1046,65	672,92

Finalment, es pot concloure que la cartera més efectiva, almenys en aquest període, hagués estat la que obeís el patró de decisió del mètode *NN* i seguint interès *SIMPLE*. En tots els casos el valor acumulat de la cartera global ha estat superior atenent el cas de cartera simple. No obstant, seguint qualsevol dels mètodes, el valor acumulat en cartera ha superat al que s'hagués obtingut mantenint posició ja que el preu del Bitcoin ha caigut 40% en el període analitzat. Cal tenir present, que per a la cartera del tipus simple, s'ha de disposar sempre de diners extra, ja que cada operació es realitza amb 1000€ independents de les operacions anteriors. Tot i que

s'hagués obtingut un millor resultat en cartera, el fet d'haver de disposar de 1000€ en cada operació dificultaria la tasca d'automatització i la disponibilitat de recursos.

7. CONCLUSIONS

Vèncer el propi mercat (superar el rendiment d'un actiu referència o índex en un període determinat) sempre serà una tasca àrdua i gairebé inassolible. Els bots de *trading* que actualment funcionen estan en mans de grans corporacions i el seu desenvolupament involucra molts treballadors altament qualificats en disciplines relacionades amb l'estadística, les matemàtiques, la computació i les finances. Paradoxalment, si un bot fos molt bo i estigués a l'abast de tothom, deixaria de ser bo. Atès que totes les operacions realitzades involucrarien tot el volum de negociació possible en cada instant de temps, provocarien un problema infinit i irresoluble en l'elecció de moments de compra i venda. Es pot afirmar, doncs, que un bot de *trading* només pot arribar a funcionar si l'utilitza un nombre petit de persones, un baix volum de negociació i no influeix directament en la cotització real mitjançant les seves ordres.

Durant tota la elaboració del treball, s'ha vist com obtenir dades en temps real i com tractar-les. S'han elaborat tres mètodes de construcció de prediccions per alimentar les decisions d'aquest bot. El resultat final no deixa de ser orientatiu, s'ha realitzat en base a 1440 operacions corresponents a 60 dies seguits d'operativa. Per tal de validar si un bot com aquest pot arribar a ser funcional caldria realitzar simulacions amb un nombre d'observacions molt superior, probablement, de més de dos anys i desenes de milers d'observacions. La potència de còmput ha estat la principal limitació per no verificar un conjunt molt superior d'observacions.

- De les tres metodologies usades per alimentar la decisòria del bot hi ha hagut una que ha estat clarament guanyadora, la de les xarxes neuronals. Si es pren només el cas amb interès compost, aquest ha obtingut uns resultats molt positius en quant a evolució del valor de la cartera en euros. De fet, és l'única que ha aconseguit una rendibilitat positiva a final de tot el període i, més important, una rendibilitat mitjana superior a 0. Les prediccions realitzades amb els tres mètodes han estat variables.
- Amb les altres dues metodologies no s'han pogut obtenir conclusions tant positives. Amb *ARIMA*, la pèrdua mitjana i final no ha estat tant elevada com amb *l'Extreme Learning Machine*, però no s'ha d'oblidar que a fi de comptes, el que ha generat és una pèrdua. Cal fer però, un matís important. Degut a la naturalesa del bot de *trading*, aquest no distingeix entre euros i Bitcoins, ja que l'únic objectiu que te és augmentar el nombre d'euros respecte Bitcoin. Al bot tant li fa que el mercat baixi o pugi, ja que pot fer operacions de tipus *CALL* i de tipus *PUT*, amb lo qual pot generar guanys en ambdós escenaris.
- Tot i que els resultats obtinguts amb els mètodes *ARIMA* i *ELM* poden semblar molt dolents, depenent de com s'analitzi, no ho són tant. Suposem el cas d'una persona que tenia 1000€ a l'inici del període (6 d'abril) i decideix invertir-los en el bot perquè

aquest els gestioni de forma activa i automàtica. Seguint els tres mètodes, la persona en qüestió hauria guanyat un 4,6% amb NN, i hagués perdut un 10,55% i un 34,3% amb els mètodes *ARIMA* i *ELM* respectivament. Suposem que una altra persona enlloc de tenir 1000€ en efectiu, té un valor en Bitcoin equivalent en 1000€ a data d'inici (6 d'abril). Si aquesta persona hagués mantingut la seva posició sense operar en cap bot, seguiria tenint el mateix nombre de Bitcoins, però el seu valor en euros seria molt inferior. El preu del Bitcoin des de l'inici del període analitzat fins al final, ha caigut des de 49127€ fins a 29598€, una caiguda d'un 40% aproximadament. Per tant, per aquesta segona persona, invertir en el bot seguint qualsevol dels tres mètodes hagués estat més rentable que mantenir la posició. En el millors dels casos, un +5% en contra d'un -40% representa un guany brutal en aquest període de 2 mesos. Si prenem el cas de l'interès simple, els resultats obtinguts han estat superiors als de la cartera amb interès simple. S'han obtingut unes rendibilitats en tots els casos superiors a les del compost, del -1,6% , +11,4% , -29,58% per *ARIMA*, *NN* i *ELM* respectivament. En cas que s'implementés el bot per operar amb diners reals, s'hauria de tenir present aquesta diferència i a priori no hi hauria cap raó per decidir apostar per l'interès compost.

- Podem deduir que s'obtenen diferents resultats, no només degut al valor acumulat en cartera sinó també per totes les proves estadístiques realitzades posteriorment. Degut al valor de correlació negatiu entre la sèrie de rendibilitats obtingudes mitjançant les *NN* amb el Bitcoin, podem arribar a pensar que en períodes de temps baixistes, és el model òptim amb el que operar. En quant als altres dos, les prediccions de tendència del mètode *Extreme Learning Machine* no acaba de funcionar del tot bé, per tant es podria descartar. Les prediccions calculades amb els models dinàmics *ARIMA*, pot ser útil en períodes de temps alcista atès que la correlació del valor en cartera i la cotització ha estat positiu.

Per acabar, la implementació del bot seria el següent pas que, segurament, realitzaré en un futur quan acabi d'optimitzar tots els algorismes. Els canvis que vull afegir van més enllà dels mètodes que he aplicat fins ara. Passen per valorar models de sèries temporals multivariants que valorin, per exemple, la variància en cada interval horari, així com altres indicadors típicament usats en l'anàlisi tècnic com el *RSI*, i alguns altres oscil·ladors. En definitiva, contemplar mètodes de *Machine Learning* des de un enfoc multivariant per tal de poder millorar les prediccions i tenir una perspectiva més global de cara a optimitzar el bot. Per exemple algorismes de *clustering* aplicats sobre sèries temporals crues com l'algorisme *SOTA*, que és un híbrid entre una xarxa neuronal no supervisada i un algorisme jeràrquic divisiu amb estructura d'arbre binari, o els models *VARMA*, que a diferència dels models *ARIMA* involucra més d'una variable. En segon lloc, i segurament el més important, caldrà depurar la decisòria operativa. Seleccionar aquells moments de temps en els que sigui més "fiable" operar i no realitzar operacions a totes hores. En el cas d'aquest treball, degut al temps limitat d'execució,

no s'ha pogut indagar més en aquest camí tot i que no descarto reprendre-ho de cara a un TFM o, fins i tot, en la creació d'una *start-up* en un futur, en cas de que ho consideri viable. Com s'ha vist, el més important és l'esperança matemàtica, i és millor encertar una gran operació que encertar-ne 5 de petites. El filtratge i depuració per decidir quan operar són tant o més importants que els propis algorismes dissenyats de cara a realitzar operacions.

El Bitcoin ha estat l'actiu escollit per a l'elaboració del treball per tot un seguit de motius exposats a l'inici. El període temporal escollit sobre el que s'han realitzat les operacions ha estat l'horari. En un futur es podria valorar si els mètodes funcionen millor amb dades de naturalesa diària, bihorària o inclús setmanal. A més, tot i que només s'ha aplicat l'operativa en aquest actiu, l'estratègia no està limitada als criptoactius i podria fàcilment replicar-se per qualsevol producte financer. La complicació estaria en poder automatitzar l'script d'R perquè accedís al *broker* des del que volguéssim realitzar les operacions.

El proper pas a seguir per implementar el bot seria automatitzar l'*script* i introduir funcions que ja es contemplen en la llibreria *rgdax*, per tal de poder efectuar operacions sense intervenció humana. Els *scripts* d'R es poden automatitzar mitjançant l'eina de Windows de *task scheduler*. Aquest procés es realitzarà en un futur quan es tingui certesa de l'efectivitat del bot

8. BIBLIOGRAFIA

Referències

- Arce, R., & Mahía, R. (2003). Modelos Arima. *Programa CITUS: Técnicas de Variables Financieras*.
- Ding, S., Xu, X., & Nie, R. (2014). Extreme learning machine and its applications. *Neural Computing and Applications*, 25(3), 549-556.
- Gilbert, H., & Handschuh, H. (2003, August). Security analysis of SHA-256 and sisters. In *International workshop on selected areas in cryptography* (pp. 175-193). Springer, Berlin, Heidelberg.
- Koskela, T., Lehtokangas, M., Saarinen, J., & Kaski, K. (1996, September). Time series prediction with multilayer perceptron, FIR and Elman neural networks. In *Proceedings of the World Congress on Neural Networks* (pp. 491-496). INNS Press San Diego, USA.
- López Romero, E. (2009, August) “El 90 por ciento de los billetes en Estados Unidos tienen restos de cocaína”. Infosalus.com <https://www.infosalus.com/actualidad/noticia-90-ciento-billetes-estados-unidos-tienen-restos-cocaina-20090817194014.html>
- Lunneborg, C. E. (1985). Estimating the correlation coefficient: The bootstrap approach. *Psychological bulletin*, 98(1), 209.
- Segendorf, B. (2014). What is Bitcoin. *Sveriges riksbank economic review*, 2.
- Ullibarri Galparsoro, L., & Pita Fernández, S. (1999). Medidas de concordancia: el índice de Kappa. *Cad Aten Primaria*, 6, 169-171.

Pàgines web consultades

Admiral Markets (2021, 21 January) ¿Qué son los Robots de Trading Automáticos?, ¿funcionan?

<https://admiralmarkets.com/es/education/articles/automated-trading/robots-forex>

bit2me academy: Ventajas de Bitcoin

<https://academy.bit2me.com/ventajas-Bitcoin/>

bit2me academy: ¿Cómo funciona el Blockchain- Cadena de Bloques?

<https://academy.bit2me.com/como-funciona-blockchain-cadena-de-bloques/>

bit2me academy: ¿Qué es SHA-256?

<https://academy.bit2me.com/sha256-algoritmo-Bitcoin/>

Carlos J. Gil Bellosta (2018, 22 April) R para profesionales de los datos: una introducción /shiny

https://www.datanalytics.com/libro_r/shiny.html

Chaitra H. Nagaraja (2014) Computational Statistics with R

<https://www.sciencedirect.com/topics/mathematics/resampling-method>

Cointelegraph: Blockchain: Qué es y como funciona, Guía para principiantes

<https://es.cointelegraph.com/Bitcoin-for-beginners/how-does-blockchain-work>

Dr. Shirin Glander (2018, 5 novembre) ‘How do neural nets learn?’ A step by step explanation using the H2O Deep Learning algorithm

<https://www.r-bloggers.com/2018/11/how-do-neural-nets-learn-a-step-by-step-explanation-using-the-h2o-deep-learning-algorithm/>

Expansión(2020): PIB- Producto Interior Bruto

<https://datosmacro.expansion.com/pib>

finnstats (2021, 9 April) Deep Neural Network in R

<https://www.r-bloggers.com/2021/04/deep-neural-network-in-r/>

Joaquin Amat (2016, May) Resampling: test de permutación, simulación de Monte Carlo y Bootstrapping

https://www.cienciadedatos.net/documentos/23_resampling_test_permutacion_simulacion_de_monte_carlo_bootstrapping#Bootstrapping_para_datos_pareados

Kernal Edem (2020, 29 May) Not so quick introduction about what is ELM. Is it really an innovation or just an iteration

?<https://towardsdatascience.com/introduction-to-extreme-learning-machines-c020020ff82b>

Nikos (2019, 16 June) Tutorial for the nnfor R Package

<https://kourentzes.com/forecasting/2019/01/16/tutorial-for-the-nnfor-r-package/>

Satoshi Nakamoto 2009 Bitcoin: A Peer-to-Peer Electronic Cash System <https://Bitcoin.org/es/Bitcoin-documento>

9. ANNEX

CODI BACKTESTING

LLIBRERIES

```

```{r}
library(stringi)
library(httr)
library(xts)
library(TTR)
library(dplyr)
library(plyr)
library(zoo)
library(forecast)
library(fBasics)
require(devtools)
library(tsviz)
library(nnfor)
install_github("rogergarciamarfany/rgdax-1",ref="dev")
library(rgdax)
library(randomForest)
library(descr)
library(lubridate)
library(gtools)
library(descr)
library(pastecs)
library(plotly)
library(irr)
library(kableExtra)
#detach("package:rgdax", unload = TRUE)
#install_github("rogergarciamarfany/rgdax",ref="dev")
#install_github("rogergarciamarfany/rgdax-1",ref="dev")
library(bootstrap)
library(boot)
library(ggplot2)
library(rgdax)
#####
```

```

FUNCIÓ OBTENIR PREUS

```

```{r eval=FALSE, include=FALSE}
Function
coinbaseGET <- function (product_id, start, end, granularity) {
 # request url
 req.url <- paste0("https://api.pro.coinbase.com/products/", product_id, "/candles")
 req.url
 # fetching the data ourselves - the return

```



```

res <- httr::GET(url = req.url,
 query = list(start = start, end = end,
 granularity = granularity))
res <- as.data.frame(t(matrix(unlist(httr::content(res)), nrow = 6)))
res[["V1"]] <- as.POSIXct(.subset2(res,1L), origin="1970-01-01")
res
}
Variables3
product_id = "BTC-EUR"
granularity = 3600
start_dates <- Sys.time() - 86400*(1:100) # considering the last 10 days
end_dates <- Sys.time() - 86400*(0:90)
Looping
res <- lapply(seq_along(end_dates),
 function(k) coinbaseGET(product_id, start_dates[k], end_dates[k], granularity))
res <- do.call(rbind, res)
bt1 <- res[,c(1,4)]
bt1_ <- bt1[nrow(bt1):1,c(1,2)]
```

```

Filtratge variables interessants

```

```{r}
res2 <- res[nrow(res):1,]
bt <- res2[,c(1,4)]
bitc <- res2[,4]
```

```

VECTOR VALORS

```

```{r}
comb <- 0:6
permutaciones <- permutations(6, 2, comb)
valors <- cbind(p=permutaciones[,1], d = rep(1,nrow(permutaciones)), q=permutaciones[,2])
```

```

METODOLOGIES

ELM PACO

```

```{r}
ELMTESTING <- function(bitc,n,d){
k <- data.frame("time" = rep(0,n), "valor" = rep(0,n), "valorn1" = rep(0,n), "nnpred" =
rep(0,n), "prediccion" = rep(0,n))
for(i in 1:n){
 bitc1 <- bitc[(1+i):(d+i)]
 b <- elm(ts(bitc1))
 a <- forecast(b,h=1)
 j <- as.numeric(a$mean)

```

```

k[i,"valor"]<-bitc[d+1+i-1]
k[i,"valorn1"]<-bitc[d+1+i]
k[i,"nnpred"]<-j
k[i,"time"]<-as.character(ymd_hms(bt1[d+i,1]))
 if((j -bitc[d+i])*(bitc[d+1+i]- bitc[d+i]) >0){
 k[i,"prediccio"]<-1
 }
print(i)
}
write.csv2(k, "ELMTESTING.csv")
}
...

```

### ## NNTESTING ##

```

```{r}
NEURALPRED<-function(bitc,x){
serie<-ts(bitc[1:(length(bitc))])
fit2 <- mlp(serie, hd = c(10,5),reps=20,difforder=0)
frc2<-forecast(fit2,h=x)
frc2
}

NNTESTING<-function(bitc,n,d){
k<-data.frame("time" = rep(0,n),"valor" = rep(0,n), "valorn1" = rep(0,n), "nnpred" =
rep(0,n),"prediccio"= rep(0,n))
for(i in 1:n){
  bitc1<-bitc[(1+i):(d+i)]
  a<-NEURALPRED(ts(bitc1),1)
  a<-as.numeric(a$mean)
k[i,"valor"]<-bitc[d+1+i-1]
k[i,"valorn1"]<-bitc[d+1+i]
k[i,"nnpred"]<-a
k[i,"time"]<-as.character(ymd_hms(bt1[d+i,1]))
  if((a -bitc[d+i])*(bitc[d+1+i]- bitc[d+i]) >0){
    k[i,"prediccio"]<-1
  }
print(i)
}
write.csv2(k, "NNTESTING.csv")
}

...

```

ARIMAS

```

```{r}
ARIMAS2<-function(bitc, valors, x) {
p<-c()
d<-c()
q<-c()
eqm<-c()
eam<-c()
epam<-c()
encert<-c()
pred<-c()
reqm<-c()
for(i in 1:nrow(valors)){
 # EX-POST
 serie<-bitc[1:(length(bitc)-x)]
 model2<-arima(serie,as.vector(valors[i,]),method="ML")
 lserie2<-predict(model2, n.ahead =x)
 #ts.plot(bitc,lserie2$pred,col=1:2)
 errors<-bitc[(length(bitc)):length(bitc)][1:x]-lserie2$pred
 eqm<- c(eqm,sum(errors*errors)/x)
 reqm<-c(reqm,eqm^(.5))
 eam<-c(eam,sum(abs(errors))/x)
 epam<-c(epam,sum(abs(errors)/abs(bitc[(length(bitc)-x):length(bitc)][1:x]))/x)
 pred<-c(pred,lserie2[1])
 lserie2<-as.vector(lserie2)
 t<-0
 for(j in 1:x){
 if((lserie2$pred[j]-bitc[length(bitc)-1])*(bitc[length(bitc)]-bitc[length(bitc)-1])>=0){
 t<-t+1
 }
 }
 encert<-c(encert,round(t,3))
}
(a<-cbind(valors,eqm,reqm,eam,epam,encert,pred))
a<-as.data.frame(a)
a$encert<-as.numeric(a$encert)
b<-a[which(a$encert==1),]
b<-as.data.frame(b)
b$eam<-as.numeric(b$eam)
b1<-b[order(b$eam),]
if(nrow(b1) == 0){
 a1<-as.data.frame(a)
 a1$eam<-as.numeric(a1$eam)
 a1<-a[order(a1$eam),]
 b<-a1[1,]
}
return(b[1,])
}
```

```

ARIMAS TESTING

```

```{r}
ARIMATESTING<-function(bitc,bt1,n,d){
f<-data.frame("time" = rep(0,n),"valor" = rep(0,n), "valorn1" = rep(0,n), "arimapred" =
rep(0,n),"prediccio"= rep(0,n))
x<-1
for(i in 1:n){
bitc2<-bitc[(1+i):(d+i)]
s<-ARIMAS2(bitc2, valors,1)
model2<-
arima(bitc2,order=c(as.numeric(unlist(s[, "p"])),as.numeric(unlist(s[, "d"])),as.numeric(unlist(s
[, "q"]))),method="ML")
lserie2<-predict(model2, n.ahead =1)
pr<-as.numeric(lserie2$pred)
f[i,"valor"]<-bitc[d+1+i-1]
f[i,"valorn1"]<-bitc[d+1+i]
f[i,"arimapred"]<-pr
f[i,"time"]<-as.character(ymd_hms(bt1[d+i,1]))
 if((pr -bitc[d+i])*(bitc[d+1+i]- bitc[d+i]) >0){
 f[i,"prediccio"]<-1
 }
}
print(i)
}
write.csv2(f, "ARIMATESTING.csv")
}
```
```{r}

bitc2<-bitc[500:length(bitc)]
bt1<-bt[500:nrow(bt),]

```

**##### GENERACIÓ 3 CSV #####**

```

ARIMATESTING(bitc2,bt1,1000,300)
NNTESTING(bitc2,1000,300)
ELMTESTING(bitc2, 1000,300)

```

```

```

```

ANALYSIS

```

```{r}
library(readxl)
library(plotly)
datos<-read_excel("E:/universitat/quart/TFG/1441_150/Compleat.xlsx")
datos$PRED<-datos$aprediccio+datos$nprediccio + datos$eprediccio
freq(datos$PRED)
fig <- plot_ly(x = datos$time, y=datos$valor, type = 'scatter',name= 'Cotització',mode =
'lines', yaxis= "BTC",)

```

```

fig <- fig %>% add_trace(y = ~datos$arimaped, name = 'ARIMA', mode = 'lines+markers')
fig <- fig %>% add_trace(y = ~datos$nnpred, name = 'NN', mode = 'lines+markers')
fig <- fig %>% add_trace(y = ~ datos$elpred, name = 'ELM', mode = 'lines+markers')
fig
```



```

```{r}
fig <- plot_ly(alpha = 0.6)
fig <- fig %>% add_histogram(x = ~datos$`Biaix ARIMA`, name = "ARIMA")
fig <- fig %>% add_histogram(x = ~ datos$`Biaix NN`, name = "NN" )
fig <- fig %>% add_histogram(x = ~ datos$`Biaix ELM`, name = "ELM", colour = "grey")
fig <- fig %>% layout(barmode = "overlay")
fig
library(descr)
library(pastecs)
kable(round(stat.desc(datos$`Biaix ARIMA`),5))
kable(round(stat.desc(datos$`Biaix NN`),5))
kable(round(stat.desc(datos$`Biaix ELM`),5))
```

```



```

```{r}
freq(datos$PRED,plot=F)
encerts<-c(0,1,2,3)
percentages<-c(14.3,34.35,35.46,15.89)
fig <- plot_ly(
  x = c(0,1,2,3),
  y = c(14.3,34.35,35.46,15.89),
  name = "Encerts models",
  type = "bar", text= c("14.3","34.35","35.46","15.89"), textposition = 'auto',
  marker = list(color = 'rgb(158,202,225)',
  line = list(color = 'rgb(8,48,107)', width = 1.5)))

fig <- fig %>% layout(title = "Encerts models",
  xaxis = list(title = ""),
  yaxis = list(title = ""))
fig
```

```



```

```{r}
datos$tend<-rep(0,nrow(datos))
for(i in 1:nrow(datos)){
  if(datos[i,"valorn1"]> datos[i,"valor"]){
    datos[i,"tend"]<-1
  }
}
freq(datos$tend)
```

```



```

```{r}
gr0<-datos[which(datos$tend==0),]
gr1<-datos[which(datos$tend ==1),]

```


```

```

freq(gr0$PRED,plot=F)
freq(gr1$PRED,plot=F)
```


```{r}



```

y<-c(17.2,36.06,34.4,12.34)
y2<-c(11.39,32.64,36.53,19.44)
x<-c("0","1","2","3")
text <- c('27% market share', '24% market share', '19% market share', '12% market share')
data <- data.frame(x, y, y2, text)
fig <- data %>% plot_ly()
fig <- fig %>% add_trace(x = ~x, y = ~y, type = 'bar', name = "baixada",
  text = y, textposition = 'auto',
  marker = list(color = 'rgb(255,51,51)',
    line = list(color = 'rgb(51,0,0)', width = 1.5)))
fig <- fig %>% add_trace(x = ~x, y = ~y2, type = 'bar', name = "pujada",
  text = y2, textposition = 'auto',
  marker = list(color = 'rgb(153,255,153)',
    line = list(color = 'rgb(51,0,0)', width = 1.5)))
fig <- fig %>% layout(title = "% Encerts",
  bargroup = 'group',
  xaxis = list(title = ""),
  yaxis = list(title = ""))
fig
```

```


```

Índex de KAPPA

```

```{r}
Kap<-datos[,c("apredicció","npredicció","epredicció")]
ARi<-Kap$apredicció
Nni<-Kap$npredicció
ELi<-Kap$epredicció
kappam.fleiss(Kap, detail =T)
kappam.fleiss(Kap[,c(2,3)], detail =T)
```

```

Valor acumulat

```

```{r}
datos$Aricon<-rep("V",nrow(datos))
datos$NNcon<-rep("V", nrow(datos))
datos$ELMcon<-rep("V", nrow(datos))
for(i in 1:nrow(datos)){
 if(datos[i,"arimapred"] > datos[i, "valor"]){
 datos[i,"Aricon"]<- "C" }
 if(datos[i,"nnpred"] > datos[i, "valor"]){
 datos[i,"NNcon"]<- "C"
 }
 if(datos[i,"elmpred"] > datos[i, "valor"]){
 datos[i,"ELMcon"]<- "C"
 }
}

```

```

}
}
freq(datos$Aricon,plot=F)
freq(datos$NNcon,plot=F)
freq(datos$ELMcon,plot=F)
datos$Aricon<-as.factor(datos$Aricon)
```

```

ARIMA

```

```{r}
t<-rep(1,nrow(datos))
for(i in 1:(nrow(datos)-1)){
if(datos[i,"Aricon"] == "C"){

 t[i+1]<- t[i]*(1+(datos[i,"valorn1"]-datos[i,"valor"])/datos[i,"valor"])
}
if(datos[i,"Aricon"] == "V"){
 t[i+1]<- t[i]*(1+(datos[i,"valor"]-datos[i,"valorn1"])/datos[i,"valor"])
}
}
ARIMAartificial<-as.numeric(t)
```

```

NN

```

```{r}
t<-rep(1,nrow(datos))
for(i in 1:(nrow(datos)-1)){
if(datos[i,"NNcon"] == "C"){
 t[i+1]<- t[i]*(1+(datos[i,"valorn1"]-datos[i,"valor"])/datos[i,"valor"])
}
if(datos[i,"NNcon"] == "V"){
 t[i+1]<- t[i]*(1+(datos[i,"valor"]-datos[i,"valorn1"])/datos[i,"valor"])
}
}
NeuronalARt<-as.numeric(t)
```

```

ELM

```

```{r}
t<-rep(1,nrow(datos))
for(i in 1:(nrow(datos)-1)){
if(datos[i,"ELMcon"] == "C"){
 t[i+1]<- t[i]*(1+(datos[i,"valorn1"]-datos[i,"valor"])/datos[i,"valor"])
}
if(datos[i,"ELMcon"] == "V"){
 t[i+1]<- t[i]*(1+(datos[i,"valor"]-datos[i,"valorn1"])/datos[i,"valor"])
}
}
Extremelearning<-as.numeric(t)

```

```

ARIMAartificial[length(ARIMAartificial)]*1000
NeuronalARt[length(NeuronalARt)]*1000
Extremelearning[length(Extremelearning)]*1000

...
```{r}
fig <- plot_ly(x = datos$time, y=ARIMAartificial*1000, type = 'scatter',name=
'ARIMAARTIFICIAL',mode = 'lines', yaxis= "BTC", )
fig <- fig %>% add_trace(y = ~NeuronalARt*1000, name = 'NEURALNETWORK', mode =
'lines+markers')
fig <- fig %>% add_trace(y = ~Extremelearning*1000, name =
'EXTREMELEARNINGMACHINE', mode = 'lines+markers')
fig
...
```{r}
a<-round((stat.desc(ARIMAartificial*1000)),2)
b<-round((stat.desc(NeuronalARt*1000)),2)
c<-round((stat.desc(Extremelearning*1000)),2)
kable(a)
kable(b)
kable(c)
...

```

## ## Prova permutacions

```

```{r}
B<-ARIMAartificial*1000
MD<-NeuronalARt*1000
BP<-Extremelearning*1000
data<-data.frame(B,MD,BP)
...

```

CAS NN VS ARIMA

```

```{r}
my_graph <- ggplot(data,aes(x = B, y = MD)) +
 geom_point() +
 stat_smooth(method = "lm",
 col = "blue",
 se = FALSE,
 size = 1) + xlab("ARIMA") +ylab("Neural Network")+
 geom_smooth()+theme_minimal()
my_graph
...
```{r}
r<-cor(B,MD)
nperm<-19999
set.seed(15)
r.perm <- replicate(nperm,{
cor(B, sample(MD))

```



```

}
)
p.valor <- (sum(abs(r.perm) > abs(r)) + 1) / (nperm + 1)
p.valor``
Observem un p.valor inferior al nivell de significació,
```{r}
nboot<-10000
corStat = function(xy, i)
 { xy.boot = xy[i,]
 r.boot = cor(xy.boot[,1], xy.boot[,2])
 aux = 1 - r.boot * r.boot
 var.r = aux * aux / (nrow(xy) - 3)
 return(c(r.boot, var.r))
 }
}
set.seed(1234)
r.boots = boot(data = data, statistic = corStat, R = nboot)
mean(r.boots$t[,1])
```

```

CAS ARIMA VS ELM

```

```{r}
my_graph <- ggplot(data,aes(x = B, y = BP)) +
 geom_point() +
 stat_smooth(method = "lm",
 col = "red",
 se = FALSE,
 size = 1) + xlab("ARIMA") + ylab("Extreme learning machine")+
 geom_smooth()+theme_minimal()
my_graph
```
```{r}
r<-cor(B,BP)
nperm<-19999
set.seed(15)
r.perm <- replicate(nperm,{
 cor(B, sample(BP))
})
)
p.valor <- (sum(abs(r.perm) > abs(r)) + 1) / (nperm + 1)
p.valor
```

```

Observem un p.valor inferior al nivell de significació,

```

``{r}
data
nboot<-10000
corStat = function(xy, i)
  { xy.boot = xy[i,]
    r.boot = cor(xy.boot[,1], xy.boot[,3])
    aux = 1 - r.boot * r.boot

```

```

var.r = aux * aux / (nrow(xy) - 3)
return(c(r.boot, var.r))
}
set.seed(1234)
r.boots = boot(data = data, statistic = corStat, R = nboot)
mean(r.boots$t[,1])
```

```

## ## CAS NN vs ELM

```

```{r}
my_graph <- ggplot(data,aes(x = MD, y = BP)) +
  geom_point() +
  stat_smooth(method = "lm",
    col = "green",
    se = FALSE,
    size = 1) + xlab("Neural Networkk") + ylab("Extreme learning machine")+
  geom_smooth()+theme_minimal()
my_graph
```

```

```

```{r}
r<-cor(MD,BP)
nperm<-19999
set.seed(15)
r.perm <- replicate(nperm,{
  cor(B, sample(BP))
})
p.valor <- (sum(abs(r.perm) > abs(r)) + 1) / (nperm + 1)
p.valor
```

```

Observem un p.valor inferior al nivell de significació,

```

```{r}
datanboot<-10000
corStat = function(xy, i)
  { xy.boot = xy[i,]
  r.boot = cor(xy.boot[,2], xy.boot[,3])
  aux = 1 - r.boot * r.boot
  var.r = aux * aux / (nrow(xy) - 3)
  return(c(r.boot, var.r))
}
set.seed(1234)
r.boots = boot(data = data, statistic = corStat, R = nboot)
mean(r.boots$t[,1])
```


```

```{r}
library(plotly)
bt1
length(B)
datos$ARISERIE<-as.ts(B)
datos$NNSERIE<-as.ts(MD)

```


```

```

datos$ELMSERIE<-as.ts(BP)
ARISERIE<-datos[,c("time","ARISERIE")]
NNSERIE<-datos[,c("time","NNSERIE")]
ELMSERIE<-datos[,c("time","ELMSERIE")]
a<-as.ts(ARISERIE)
library(tsviz)
prices <- crypto_prices
df<-datos[,c("time","valor","ARISERIE","NNSERIE","ELMSERIE")]
df<-as.data.frame(df)
df$time<-as.Date(df$time)
df$t<-1:nrow(df)
df$t<-as.Date(df$t)
```


TEST CORR CARTERAS VS COTITZACIÓ


```

## ## Prova permutacions

```

```{r}
B<-ARIMAartificial*1000
MD<-NeuronalARt*1000
BP<-Extremelearning*1000
Bitcoin<-datos$valor
data<-data.frame(B,MD,BP,Bitcoin)
```

```

## ### CAS NN VS ARIMA

```

```{r}
my_graph <- ggplot(data,aes(x = B, y = Bitcoin)) +
  geom_point() +
  stat_smooth(method = "lm",
    col = "blue",
    se = FALSE,
    size = 1) + xlab("ARIMA") + ylab("Bitcoin")+ geom_smooth()+theme_minimal()
my_graph
```

```

```

```{r}
r<-cor(B,Bitcoin)
nperm<-19999
set.seed(15)
r.perm <- replicate(nperm,{
  cor(B, sample(Bitcoin))
})
p.valor <- (sum(abs(r.perm) > abs(r)) + 1) / (nperm + 1)
p.valor
```

```

Observem un p.valor inferior al nivell de significació,

```

```{r}
nboot<-10000

```

```

corStat = function(xy, i)
  { xy.boot = xy[i,]
  r.boot = cor(xy.boot[,1], xy.boot[,4])
  aux = 1 - r.boot * r.boot
  var.r = aux * aux / (nrow(xy) - 3)
  return(c(r.boot, var.r))
}
set.seed(1234)
r.boots = boot(data = data, statistic = corStat, R = nboot)
mean(r.boots$t[,1])

...

```

CAS ELM vs Bitcoin

```

...{r}
my_graph <- ggplot(data,aes(x = BP, y = Bitcoin)) +
  geom_point() +
  stat_smooth(method = "lm",
    col = "red",
    se = FALSE,
    size = 1) + xlab("ARIMA") + ylab("Extreme learning machine")+
  geom_smooth()+theme_minimal()
my_graph
...

```

```

...{r}
r<-cor(B,Bitcoin)
nperm<-19999
set.seed(15)
r.perm <- replicate(nperm,{
  cor(BP, sample(Bitcoin))
})
p.valor <- (sum(abs(r.perm) > abs(r)) + 1) / (nperm + 1)
p.valor
...

```

Observem un p.valor inferior al nivell de significació,

```

...{r}
data
nboot<-10000
corStat = function(xy, i)
  { xy.boot = xy[i,]
  r.boot = cor(xy.boot[,3], xy.boot[,4])
  aux = 1 - r.boot * r.boot
  var.r = aux * aux / (nrow(xy) - 3)
  return(c(r.boot, var.r))
}
set.seed(1234)

```

```
r.boots = boot(data = data, statistic = corStat, R = nboot)
mean(r.boots$t[,1])
```

```

## ## CAS NN vs BITCOIN

```
```{r}
my_graph <- ggplot(data,aes(x = MD, y = Bitcoin)) +
  geom_point() +
  stat_smooth(method = "lm",
    col = "green",
    se = FALSE,
    size = 1) + xlab("Neural Networkk") +ylab("Extreme learning machine")+
  geom_smooth()+theme_minimal()
my_graph
```

```

```
```{r}
r<-cor(MD,Bitcoin)
nperm<-19999
set.seed(15)
r.perm <- replicate(nperm,{
  cor(B, sample(Bitcoin))
})
p.valor <- (sum(abs(r.perm) > abs(r)) + 1) / (nperm + 1)
p.valor
```

```

Observem un p.valor inferior al nivell de significació,

```
```{r}
data
nboot<-10000
corStat = function(xy, i)
  { xy.boot = xy[i,]
  r.boot = cor(xy.boot[,2], xy.boot[,4])
  aux = 1 - r.boot * r.boot
  var.r = aux * aux / (nrow(xy) - 3)
  return(c(r.boot, var.r))
}
set.seed(1234)
r.boots = boot(data = data, statistic = corStat, R = nboot)
mean(r.boots$t[,1])
```

```

```
```{r}
library(plotly)
bt1
length(B)
```

```

datos$ARISERIE<-as.ts(B)
datos$NNSERIE<-as.ts(MD)
datos$ELMSERIE<-as.ts(BP)

ARISERIE<-datos[,c("time","ARISERIE")]
NNSERIE<-datos[,c("time","NNSERIE")]
ELMSERIE<-datos[,c("time","ELMSERIE")]
a<-as.ts(ARISERIE)
library(tsviz)
prices <- crypto_prices
df<-datos[,c("time","valor","ARISERIE","NNSERIE","ELMSERIE")]
df<-as.data.frame(df)
df$time<-as.Date(df$time)
df$t<-1:nrow(df)
df$t<-as.Date(df$t)
```

```

**##### INTERES SIMPLE #####**

## **# ARIMA**

```

```{r}
t<-rep(0,nrow(datos))
for(i in 1:(nrow(datos)-1)){
  if(datos[i,"Aricon"]=="C"){
    t[i+1]<- (1000*(datos[i,"valorn1"]-datos[i,"valor"])/datos[i,"valor"])
  }
  if(datos[i,"Aricon"]=="V"){
    t[i+1]<- (1000*(datos[i,"valor"]-datos[i,"valorn1"])/datos[i,"valor"])
  }
}
ARIMAartificial<-as.numeric(t)
```

NN
```{r}
t<-rep(0,nrow(datos))
for(i in 1:(nrow(datos)-1)){
  if(datos[i,"NNcon"]=="C"){
    t[i+1]<- (1000*(datos[i,"valorn1"]-datos[i,"valor"])/datos[i,"valor"])
  }
  if(datos[i,"NNcon"]=="V"){
    t[i+1]<- (1000*(datos[i,"valor"]-datos[i,"valorn1"])/datos[i,"valor"])
  }
}

NeuronalARt<-as.numeric(t)
```

```

**# ELM**

```

```{r}
t<-rep(0,nrow(datos))
for(i in 1:(nrow(datos)-1)){
if(datos[i,"ELMcon"] == "C"){
  t[i+1]<- (1000*(datos[i,"valorn1"]-datos[i,"valor"])/datos[i,"valor"])
}
if(datos[i,"ELMcon"] == "V"){
  t[i+1]<- (1000*(datos[i,"valor"]-datos[i,"valorn1"])/datos[i,"valor"])
}
}
}
Extremelearning<-as.numeric(t)
```

```{r}
fig <- plot_ly(x = datos$time, y=ARIMAartificial, type = 'scatter',name=
'ARIMAARTIFICIAL' ,mode = 'lines', yaxis= "BTC", )
fig <- fig %>% add_trace(y = ~NeuronalARt, name = 'NEURALNETWORK', mode =
'lines+markers')
fig <- fig %>% add_trace(y = ~Extremelearning, name =
'EXTREMELEARNINGMACHINE', mode = 'lines+markers')
fig
kable(round(stat.desc(ARIMAartificial),4))
kable(round(stat.desc(NeuronalARt),4))
kable(round(stat.desc(Extremelearning),4))

```

```