
This is the **published version** of the bachelor thesis:

Monsó Cadena, Marta; Romero Sánchez, Albert tut. Prototipo de una aplicación para predecir la tendencia de los precios de las acciones utilizando inteligencia artificial. 2025. (Enginyeria de Dades)

This version is available at <https://ddd.uab.cat/record/317351>

under the terms of the  license

Prototip d'una aplicació per predir la tendència en els preus de les accions utilitzant intel·ligència artificial

Marta Monsó Cadena

Resum — Aquest treball desenvolupa un prototip d'aplicació capaç de predir la tendència a curt termini (10 dies laborables) dels preus d'accions mitjançant models d'intel·ligència artificial. S'han implementat i comparat sis algoritmes (Ridge, Random Forest, SVR, LSTM, GRU i un híbrid LSTM+XGBoost) sobre vuit sèries històriques financeres (Amazon, Google, Euro Stoxx 50, Hang Seng, IBEX 35, Indra, P&G i S&P 500) amb dades des del 1r de gener de 2020 fins el 31 de gener de 2025. Cada model s'ha avaluat amb RMSE, MAE i R^2 i s'ha generat un forecast autoregressiu per calcular el preu dels primers deu dies laborables de febrer de 2025. Els resultats mostren que el model SVR lidera en actius com Amazon i S&P 500, mentre que el model híbrid LSTM+XGBoost obté millor rendiment en Google, Euro Stoxx 50, Indra i IBEX 35. Finalment, s'ha implementat una interfície web amb Streamlit que permet seleccionar l'actiu i el model per visualitzar les prediccions i gràfics interactius de manera senzilla i eficient. Aquest prototip demostra la utilitat de combinar algoritmes seqüencials i de conjunt per a la presa de decisions financeres a curt termini.

Paraules clau — Intel·ligència artificial, Predicció de preus d'accions, Aprenentatge automàtic, Aprenentatge profund, Prototip d'aplicació, Model híbrid, Indicadors tècnics, Mètriques d'error, Sèries temporals, Decisions financeres a curt termini.

Abstract — This project develops a prototype application capable of predicting the short-term (10 working days) trend of stock prices using artificial intelligence models. Six algorithms (Ridge, Random Forest, SVR, LSTM, GRU, and a hybrid LSTM+XGBoost) were implemented and compared across eight historical financial time series (Amazon, Google, Euro Stoxx 50, Hang Seng, IBEX 35, Indra, P&G, and S&P 500), with data spanning from January 1, 2020 to January 31, 2025. Each model was evaluated using RMSE, MAE, and R^2 metrics, and an autoregressive forecast was generated to calculate prices for the first ten business days of February 2025. Results show that SVR performs best on assets like Amazon and S&P 500, while the hybrid LSTM+XGBoost model achieves superior performance on Google, Euro Stoxx 50, Indra, and IBEX 35. Finally, a web interface was developed using Streamlit, allowing users to select assets and models to view predictions and interactive plots in a simple and efficient way. This prototype demonstrates the usefulness of combining sequential and ensemble algorithms for short-term financial decision-making.

Index Terms — Artificial intelligence, Stock price prediction, Machine learning, Deep learning, Application prototype, Hybrid model, Technical indicators, Error metrics, Time series, Short-term financial decisions.



1 INTRODUCCIÓ

La intel·ligència artificial ha revolucionat la forma en què es poden analitzar i predir les tendències del mercat actualment, la qual cosa permet prendre decisions més precises i informades, ja que els sistemes d'intel·ligència artificial tenen la capacitat de definir patrons, correlacions complexes de dades i tendències que anteriorment no eren possibles d'identificar.

Exactament, en el mercat financer la presa de decisions basada en les prediccions de les tendències dels preus de les accions pot suposar un avantatge competitiu per a les empreses i per als inversors. La predicció de preus d'accions mitjançant algoritmes d'aprenentatge automàtic ajuda a determinar el valor futur de les accions d'una empresa i d'altres actius financers que cotitzen en borsa. L'objectiu principal d'aquesta predicció és obtenir guanys importants. No obstant això, anticipar el comportament del mercat borsari és una tasca complexa, ja que hi intervenen una gran varietat de factors, incloent-hi aspectes físics, psicològics i comportaments racionals i irracionals. El dinamisme, la volatilitat del mercat i els patrons de preus sovint requereixen mètodes sofisticats per identificar tendències, riscos

i oportunitats a curt termini, fent que sigui complex aconseguir una predicció amb alta precisió.

Per aquesta raó, en aquest treball de final de grau ens proposem explorar l'ús de la intel·ligència artificial en la predicció de preus d'accions i analitzar quins models ofereixen els millors resultats en aquest àmbit. La idea és dur a terme un estudi dels diferents tipus de models d'aprenentatge automàtic i d'aprenentatge profund que s'utilitzen per a aquest propòsit, avaluant la seva eficàcia i aplicabilitat en la presa de decisions financeres. A través d'aquesta anàlisi, busquem constatar la utilitat de la IA en finances i determinar quins models resulten més eficients per identificar patrons i tendències en el mercat borsari.

2 OBJECTIUS

L'objectiu final d'aquest treball de recerca és desenvolupar un prototip d'aplicació capaç de predir tendències en els preus d'accions a curt termini (10 dies) mitjançant models d'intel·ligència artificial utilitzant dades històriques per poder facilitar la identificació de riscos i oportunitats.

Abans d'arribar al prototip d'aplicació, és necessari que primer s'assoleixin una sèrie d'objectius més específics:

1. Analitzar i seleccionar els millors models per predir la tendència en els preus d'accions.
2. Recollir dades històriques de preus d'accions de diferents empreses, processar-les i netejar-les depenent de l'estat de les dades.
3. Implementar els diferents models proposats i comparar-los utilitzant les diferents dades històriques per poder seleccionar el millor model en termes de mètriques d'error i rendiment.
4. Desenvolupar una petita aplicació que integri els models implementats i que faci la predicció de les tendències de preus en una interfície senzilla per a l'usuari.

3 ESTAT DE L'ART

3.1 Evolució de Metodologies IA en la Predicció de Preus Borsaris

Durant els últims anys, la recerca en predicció de la borsa amb intel·ligència artificial ha avançat molt tant en les tècniques com en les aplicacions concretes. Diverses revisions com les de Sonkavde et al. [1] i Jiang [2] destaquen com les xarxes neuronals recurrents, *LSTM* i *GRU*, són molt útils per capturar les dependències en el temps. Tanmateix, reconeixen que calen estratègies més fortes per reduir el soroll i evitar l'*overfitting*. Al mateix temps, els models d'aprenentatge automàtic clàssics com *SVR* i *Random Forest*, aplicats a dades econòmiques o variables exògenes, com els estudis de Wong et al. [3] i d'Huang et al. [4], han demostrat que són competitiu, especialment a curt termini (fins a 5 dies), amb errors menors entre un 5 % i un 10 %. A més, els enfocaments híbrids, on un model *LSTM* captura la dinàmica seqüencial i un *XGBoost* ajusta els residus, han pogut reduir el RMSE en més d'un 15 % en comparació amb els models individuals, tot i que els processaments són més intensos i cal més temps per afinar-los [7], [8].

Finalment, incorporar dades qualitatives com l'anàlisi de sentiment de notícies i la informació de xarxes socials es mostra com una opció molt prometedora per enriquir els models tècnics, fent-los més sensibles als moviments inesperats del mercat [9], [11]. Tot aquest conjunt d'estudis obre la porta a crear *benchmarks* oberts i metodologies híbrides, buscant desenvolupar models més positius, fiables i generalitzables.

3.2 Algoritmes de Predicció

Seguint els objectius del treball, en aquest apartat es descriuen els principis de funcionament per a cadascun dels algoritmes utilitzats per predir el preu de les accions a curt termini.

3.2.1 Ridge Regression

Ridge Regression, també coneguda com la regularització L2, és una regressió lineal que serveix per reduir els errors causats per la multicolinealitat i el sobreajustament de les dades d'entrenament ja que es basa en aplicar una penalització L2 sobre els coeficients.

Aquest algoritme funciona de manera que en el moment de trobar els coeficients, no només es minimitza l'error quadràtic, sinó també la suma dels quadrats dels coeficients:

$$\min_{\beta} \left\{ \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \gamma \sum_{j=1}^n \beta_j^2 \right\}$$

- y_i : Valor real de la variable dependent per a la mostra i .
- $\hat{y}_i$: Valor predit pel model per a la mostra i .
- β_j : Coeficient del model associat a la variable j .
- γ : Paràmetre de regularització.

En fer que els coeficients siguin més petits i més propers a zero, se simplifica el model i es redueix la seva sensibilitat a les fluctuacions aleatòries o al soroll de les dades. Això fa que sigui menys probable que el model se sobreajusti i, per tant, l'ajuda a tenir un millor rendiment amb dades noves i no vistes, millorant la seva precisió i fiabilitat general.

3.2.2 Random Forest

Random Forest és un algoritme d'aprenentatge automàtic que combina el resultat de múltiples arbres de decisió per arribar a un resultat final únic.

Per garantir la baixa correlació entre els arbres i la reducció de la variància en la predicció global, cadascun d'aquests arbres és entrenat amb una mostra de mida N lleugerament diferent obtinguda mitjançant *bootstrapping* (mostreig amb reemplaçament). Així mateix, també és essencial que cada arbre només consideri un subconjunt aleatori de les variables originals en cada punt de partició.

Per fer prediccions sobre noves observacions, es genera la mitjana de les prediccions obtingudes de tots els arbres que conformen el model.

3.2.3 Support Vector Regression (SVR)

Support Vector Regression (SVR) és una versió de les màquines de vectors de suport que serveix per fer regressió. La idea és buscar una funció $f(x)=wTx+bf$ que sigui el més "plana" possible (amb pesos petits), però que al mateix temps deixi una petita franja de tolerància ϵ al seu voltant on no comptin els petits errors.

Perquè passi això, es defineix un tub ϵ -insensitive al voltant de l'hiperplà; si la predicció cau dins el tub no es penalitza, però si una dada queda fora del tub, apareixen variables flexibles $\xi_i, \xi_i \geq 0$ que mesuren la desviació.

A més, el paràmetre C controla en quina mesura es penalitzen aquests errors fora del tub. Amb el *kernel trick*, podem capturar relacions no lineals fent veure al model que les dades viuen en un espai de més dimensions sense transformar-les de debò.

3.2.4 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) és una versió millorada de la xarxa neuronal recurrent (RNN). Aquest model pot capturar dependències a llarg termini en dades seqüencials ja que introdueix una cèl·lula de memòria que manté informació durant períodes prolongats, abordant el repte d'aprendre dependències a llarg termini.

L'arquitectura *LSTM* inclou la cèl·lula de memòria, que està controlada per tres portes:

- Porta d'entrada: controla quina informació s'afegeix a la cèl·lula de memòria.
- Porta d'oblí: determina quina informació s'elimina de la cèl·lula de memòria.
- Porta de sortida: controla quina informació surt de la cèl·lula de memòria.

3.2.5 Gated Recurrent Neural Network (GRU)

L'algoritme *GRU* és una variant simplificada del model *LSTM* que combina les portes d'oblí i d'entrada en una sola porta d'actualització. Això redueix el nombre de paràmetres i accelera l'entrenament. La idea fonamental és utilitzar mecanismes de portes per actualitzar selectivament l'estat ocult en cada pas temporal, permetent retenir la informació important i descartar-ne els detalls irrelevants.

La *GRU* consta de dues portes principals:

- Porta d'actualització (z_t): Determina quanta informació de l'estat ocult anterior s'ha de conservar per al següent pas temporal.
- Porta de reinici (r_t): Decideix quina part de l'estat ocult anterior s'ha d'oblidar.

Aquestes portes permeten a la *GRU* controlar el flux d'informació de manera més eficient en comparació amb les *RNN* tradicionals, que es basen únicament en l'estat ocult.

3.2.6 Híbrid LSTM + XGBoost

En aquest model de dos passos, primer s'entrena un model *LSTM* per capturar les dinàmiques seqüencials principals de la sèrie temporal i després s'aplica *XGBoost* sobre els residus (errors) de la *LSTM* per millorar la precisió final.

1. *LSTM*: S'entrena per predir y_t i es calculen els residus.

$$r_t = y_t - \hat{y}_t^{LSTM}$$

Aquests residus recullen la part d'error que la *LSTM* no ha aconseguit explicar.

2. *XGBoost*: Primer, entrena sobre els residus r_t usant com a predictors valors històrics, prediccions *LSTM* i indicadors tècnics i després genera r_t^{XGB} per corregir l'error.
3. Predicció final: Per a un instant nou t , es calcula:

$$\hat{y}_t = \hat{y}_t^{LSTM} - \hat{r}_t^{XGB}$$

- y_t : Valor real de la variable (preu) en el temps t .
- $\hat{y}_t$: Predicció final combinada.
- $\hat{y}_t^{LSTM}$: Predicció que fa *LSTM* per al temps t .
- r_t : Residus del model *LSTM*.
- $\hat{r}_t^{XGB}$: Predicció de l'error per part d'*XGBoost*.

3.3 Mètriques

És imprescindible comptar amb diferents mètriques per a l'avaluació dels models de regressió que s'han creat per predir el preu de les accions. Necessitem mesurar quantitativament l'error entre les prediccions $\hat{y}_i$ i els valors reals

y_i , per fer-ho s'han utilitzat les tres mètriques següents: *Root Mean Squared Error (RMSE)*, *Mean Absolute Error (MAE)* i R^2 (Coeficient de determinació).

3.3.1 Root Mean Squared Error

El *RMSE* mesura l'arrel quadrada de la mitjana dels errors al quadrat entre les prediccions i els valors reals. Per això, el *RMSE* proporciona una indicació clara de la precisió global del model. També, a causa del quadrat, penalitza molt fort els errors grans i és sensible als valors atípics.

Fórmula:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$$

- n : Nombre total de mostres.
- $\hat{y}_i$: Valor predit pel model en la mostra i .
- y_i : Valor real en la mostra i .

3.3.2 Mean Absolute Error

El *MAE* és la mitjana de les diferències absolutes entre prediccions i valors reals, és a dir, representa l'error mitjà absolut que comet el model. Es considera una mesura interpretable i robusta per a la comparació de models.

Fórmula:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

3.3.3 Coeficient de determinació

El R^2 mesura la proporció de la variància total dels valors reals que explica el model i determina la qualitat del model per replicar els resultats. El coeficient mostra com de bé s'ajusta un model de regressió a un conjunt de dades, és a dir, indica el percentatge explicat pel model de regressió.

Fórmula:

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

- $\bar{y}$: Mitjana de tots els valors reals.

El R^2 adquireix un valor entre $-\infty$ i 1, sent 1 quan es considera que la predicció és perfecta, 0 quan el model no explica més variància que la mitjana simple i més petit que zero quan el model empitjora la predicció respecte a fer servir la mitjana.

4 METODOLOGIA

La intenció és que el TFG segueixi un enfocament experimental i analític, utilitzant tècniques d'aprenentatge automàtic i d'aprenentatge profund per analitzar les dades històriques i poder determinar d'aquesta manera la tendència que seguirà el preu de les accions a curt termini.

4.1 Conjunt de Dades

4.1.1 Recollida de Dades

Per dur a terme els diferents models d'intel·ligència artificial explicats en l'estat de l'art, utilitzem les dades històriques de vuit actius financers diferents que van des del 1r de gener de 2020 fins el 31 de gener de 2025. Aquests conjunts de dades contenen 7 variables diferents, les quals són: *Date*, *Price*, *Open*, *High*, *Low*, *Vol.*, *Change %*.

A continuació, es dona la informació de cada conjunt de dades i la seva gràfica corresponent:

1. [Amazon](#): És una de les empreses tecnològiques més grans del món i el seu valor a borsa reflecteix l'estat del sector tecnològic nord-americà i el consum global. *Format per 1278 files i 7 columnes.*



Fig. 1. Gràfica de l'evolució dels preus de les accions del conjunt de dades d'Amazon.

2. [Euro Stoxx 50](#): És l'índex borsari que representa les 50 principals empreses de la zona euro reflectint l'economia europea. *Format per 1306 files i 7 columnes.*



Fig. 2. Gràfica de l'evolució dels preus de les accions del conjunt de dades d'Euro Stoxx 50.

3. [Google](#) (Alphabet Inc.): És un dels gegants tecnològics mundials sent un actiu especialment útil per predir tendències generals del sector tecnològic i la confiança del mercat en la innovació. *Format per 1278 files i 7 columnes.*



Fig. 3. Gràfica de l'evolució dels preus de les accions del conjunt de dades de Google.

4. [Hang Seng](#): És l'índex de referència de la borsa de Hong Kong que reflecteix l'evolució econòmica de Xina i Àsia, en general. *Format per 1258 files i 7 columnes.*



Fig. 4. Gràfica de l'evolució dels preus de les accions del conjunt de dades de Hang Seng.

5. [IBEX 35](#): És l'índex borsari que agrupa les 35 empreses amb més liquidesa i de major capitalització de la Borsa espanyola. *Format per 1306 files i 7 columnes.*



Fig. 5. Gràfica de l'evolució dels preus de les accions del conjunt de dades d'IBEX 35.

6. [Indra](#): És una empresa tecnològica espanyola especialitzada en sistemes d'informació, defensa i transport de referència en innovació. *Format per 1303 files i 7 columnes.*



Fig. 6. Gràfica de l'evolució dels preus de les accions del conjunt de dades d'Indra.

7. [Procter & Gamble \(P&G\)](#): És una multinacional nord-americana de béns de consum. Els seus productes són presents a nivell global i el seu comportament borsari reflecteix el consum quotidià. *Format per 1278 files i 7 columnes.*



Fig. 7. Gràfica de l'evolució dels preus de les accions del conjunt de dades de Procter & Gamble.

8. [S&P500](#): És l'índex que agrupa les 500 empreses nord-americanes de major capitalització, considerat referència de la salut de l'economia dels EUA i baròmetre del mercat global de valors. *Format per 1278 files i 7 columnes.*



Fig. 8. Gràfica de l'evolució dels preus de les accions del conjunt de dades de S&P500.

4.1.2 Preprocessament de les Dades

El preprocessament de les dades és tot un conjunt d'etapes de preparació dels conjunts de dades essencial per garantir que les dades estiguin en un format apte per als models de predicció desenvolupats. D'aquesta manera, s'han desenvolupat un seguit de tasques per les quals han passat els diferents conjunts de dades anomenats anteriorment.

Com a pas inicial, s'ha realitzat un *profiling* de les dades per obtenir una visió general del conjunt, identificar valors atípics, distribucions, valors nuls i correlacions entre variables. Utilitzant *Pandas* i *NumPy* s'han tractat valors nuls, s'ha ajustat el format de *timestamp*, s'han convertit variables de tipus objecte a numèriques i s'han creat gràfiques per visualitzar l'evolució dels preus. També s'han creat les noves característiques tècniques (*EMA_7*, *EMA_40*, *MACD*, *Signal_Line*, *MACD_Histograma*, *RSI* i *ATR*) explicades seguidament que serveixen per transformar i ampliar les variables inicials. D'aquesta manera el model tindrà informació més rellevant i estructurada, es reduirà el soroll i s'exposaran patrons claus perquè aprengui més ràpidament i amb millor qualitat.

Finalment, els conjunts de dades estan formats per 13 variables: *Date*, *Open*, *High*, *Low*, *Close*, *Volume*, *EMA_7*, *EMA_40*, *MACD*, *Signal_Line*, *MACD_Hist*, *RSI*, *ATR*; sent la variable *Close* la variable objectiu dels diferents models de predicció, és a dir, la que els models tracten d'estimar.

4.1.3 Indicadors Tècnics

Els indicadors tècnics que s'han calculat i afegit als diferents conjunts de dades són els següents:

- Mitjanes Mòbils Exponencials.
- MACD, Línia de Senyal i Histograma.
- Índex de Força Relativa (*RSI*).
- Mitjana del rang veritable (*ATR*).

4.1.3.1 Mitjanes Mòbils Exponencials

La mitjana mòbil exponencial (*EMA*, *Exponential Moving Average*), que fa el seguiment del valor de l'acció, és un tipus de mitjana mòbil ponderada que atorga més importància a les dades més recents. La informació de tendència obtinguda en els gràfics de mitjanes mòbils, especialment diverses mitjanes mòbils superposades les unes a les altres, pot ajudar a identificar punts de resistència i suport i provocar decisions de compra o venda.

Al conjunt de dades s'hi han afegit dos *EMAs* amb valors mitjans de 7 i 40. La mitjana mòbil exponencial a curt termini que creua a l'alça la mitjana mòbil exponencial a llarg termini es considera un senyal alcista, mentre que el contrari acostuma a considerar-se un senyal baixista.

L'algoritme emprat per calcular les diferents *EMAs* en el projecte és el següent:

```
# Funció per calcular la mitjana mòbil exponencial (EMA)
def exponential_moving_average(data, period):
    data = np.asarray(data) # Convertir les dades a un array de NumPy
    alpha = 2 / (period + 1) # Càlcul del factor de suavitzat (alpha)
    ema = np.empty_like(data)
    ema[0] = data[0] # El primer valor de l'EMA és igual al primer valor de les dades

    # Càlcul iteratiu de l'EMA per a cada punt
    for i in range(1, len(data)):
        ema[i] = alpha * data[i] + (1 - alpha) * ema[i - 1]

    return ema
```

Fig. 9. Funció de Python per calcular la mitjana mòbil exponencial (*EMA*).

Es comença utilitzant el primer preu de la sèrie i, a continuació, cada nou valor d'*EMA* es calcula com una combinació ponderada del preu actual i de l'*EMA* anterior segons un factor que atorga més pes a les dades recents.

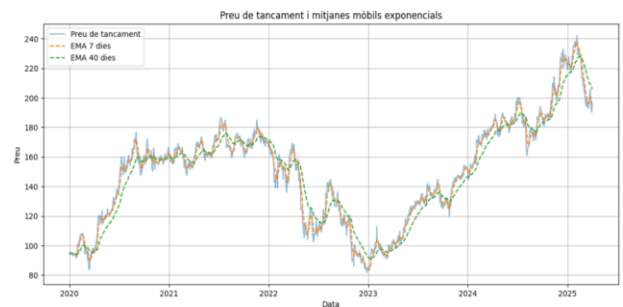


Fig. 10. Gràfica del preu de tancament i mitjanes mòbils exponencials del conjunt de dades d'Amazon.

En aquesta gràfica anterior, podem veure que durant els primers mesos del 2020, l'*EMA* de 7 dies es creua per damunt de l'*EMA* de 40 dies; això assenyalava l'inici d'una tendència a l'alça que es manté gairebé fins el 2021.

4.1.3.2 MACD, Línia de Senyal i Histograma

L'indicador *MACD* (*Moving Average Convergence Divergence*) està compost per tres components:

- Línia *MACD*.
- Línia de Senyal.
- Histograma del *MACD*.

La funció *MACD* és la diferència entre l'*EMA* de 26 períodes dels valors de tancament i l'*EMA* de 12 períodes dels valors de tancament. Aquests dos indicadors s'anomenen respectivament *EMA* lenta i *EMA* ràpida. Aquesta línia mostra la convergència o la divergència del ritme de canvi de preu a curt termini respecte del de llarg termini.

$$MACD = 12periodEMA - 26periodEMA$$

La funció de senyal és una *EMA* de 9 dies de la *MACD* i s'utilitza com a senyal d'activació que ajuda a identificar senyals de compra o de venda quan tenen lloc els encreuaments entre la línia *MACD* i la senyal. Quan *MACD* supera la senyal, la tendència de preus és alcista i a l'inrevés, quan la senyal supera el *MACD*, la tendència del mercat serà baixista. També es crea un histograma que bàsicament és la diferència entre la línia *MACD* i la línia de senyal per poder visualitzar i identificar la força i la durada d'una tendència abans que es produeixi un possible creuament de línies. Quan l'histograma és zero, indica un canvi en la tendència del mercat.

L'algorisme emprat per calcular les diferents funcions, que utilitza la funció *exponential_moving_average* explicada anteriorment per obtenir els *EMA* necessaris per calcular els diferents indicadors, és el següent:

```
# funció per calcular el MACD, la línia de senyal i l'histograma
def calculate_macd(df, short_period=12, long_period=26, signal_period=9):

    # Calcular la mitjana mòbil exponencial curta i llarga del preu de tancament
    ema_short = exponential_moving_average(df['close'], short_period)
    ema_long = exponential_moving_average(df['close'], long_period)

    # Calcular el MACD com la diferència entre les dues EMAs
    macd = ema_short - ema_long

    # Calcular la línia de senyal (mitjana suau del MACD)
    signal_line = exponential_moving_average(macd, signal_period)

    # Calcular l'histograma com la diferència entre el MACD i la línia de senyal
    macd_hist = macd - signal_line

    # Afegir les columnes resultants al DataFrame
    df['MACD'] = macd
    df['signal_line'] = signal_line
    df['MACD_hist'] = macd_hist

    return df
```

Fig. 11. Funció de Python per calcular la *MACD*, la línia de senyal i l'histograma del *MACD*.

Si ens fixem en la gràfica, els creuaments entre la línia *MACD* i la línia de senyal, reforçats per l'amplitud de l'histograma, indiquen els punts de gir de la tendència en el preu de les accions i la seva intensitat en el mercat.

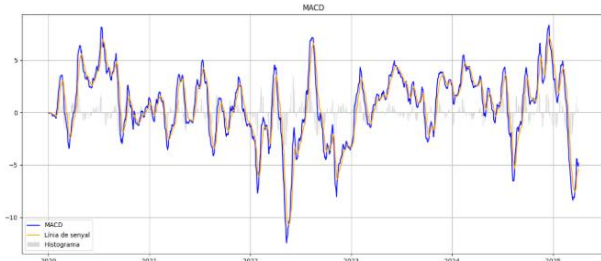


Fig. 12. Gràfica de l'indicador *MACD* generat per al conjunt de dades d'Amazon.

4.1.3.3 Índex de Força Relativa (RSI)

L'*RSI* (*Relative Strength Index*) és un indicador tècnic d'impuls (*momentum*) que mesura la velocitat i el canvi dels moviments del preu; mesura la magnitud de les pujades en vers les baixades en un període de temps de 14 dies, proporcionant un valor entre 0 i 100. Es considera com a zona de sobreventa la compresa entre 0 i 30, com a zona de sobrecompra la compresa entre 70 i 100, i com a zona neutra els nivells entre 30 i 70.

Si el preu de les accions ha baixat més del que ha pujat durant el període que contempla l'indicador (14 dies en el nostre cas), l'*RSI* estarà per sota de 50. Si el preu ha pujat més del que ha baixat, l'*RSI* estarà per sobre de 50.

S'utilitza la següent funció per calcular el *RSI*:

```
# Calcular els guanys i les pèrdues: els guanys són les variacions positives, les pèrdues són les negatives (en valor absolut)
gain = delta.clip(lower=0)
loss = -delta.clip(upper=0)

# Calcular la mitjana exponencial dels guanys i les pèrdues
avg_gain = gain.ewm(alpha=1/period, adjust=False, min_periods=period).mean()
avg_loss = loss.ewm(alpha=1/period, adjust=False, min_periods=period).mean()

# Calcular la força relativa (RS)
rs = avg_gain / avg_loss

# Calcular l'RSI
rsi = 100 - (100 / (1 + rs))

df['RSI'] = rsi

return df
```

Fig. 13. Funció de Python per calcular l'indicador *RSI*.

Es calcula la diferència diària de tancament, separant pujades (guanys) i baixades (pèrdues). A cadascuna d'aquestes sèries s'aplica una mitjana mòbil exponencial en un període de 14 dies per obtenir la mitjana de guanys i la de pèrdues. El seu quocient defineix la Força Relativa (*RS*), que es transforma a l'escala 0-100 mitjançant $100 - 100/(1+RS)$.

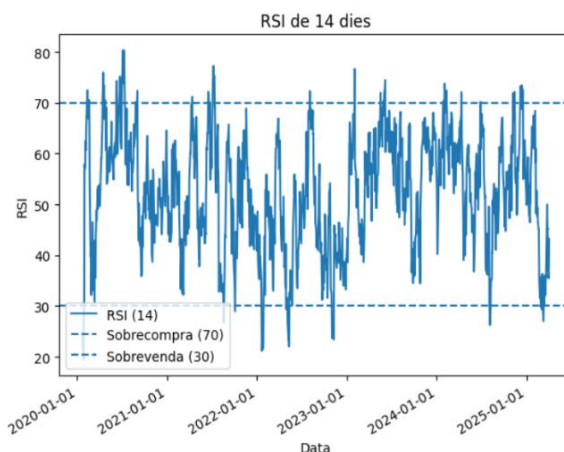


Fig. 14. Gràfica de l'indicador *RSI* generat per al conjunt de dades d'Amazon.

En aquesta gràfica veiem que majoritàriament el *RSI* oscil·la entre 40 i 60, indicant fases de consolidació amb relativament pocs extrems.

4.1.3.4 Mitjana del rang veritable (ATR)

La Mitjana del rang veritable (*ATR*) és una mesura de la volatilitat proposada per Welles Wilder, que en lloc de mirar només la diferència entre màxim i mínim de cada període, té en compte també els buits (*gaps*) respecte el tancament anterior.

Un valor de l'*ATR* més alt correspon a una volatilitat més gran del mercat subjacent i un valor de l'*ATR* més baix

indica el contrari. Per tant, l'ATR indica quan els moviments dels preus es podrien tornar més o menys esporàdics a mesura que la volatilitat augmenta o disminueix.

Primer, es calcula la True Range de cada dia com la màxima de $(High-Low)$, $|High-Close_{-1}|$ i $|Low-Close_{-1}|$, després s'obté el primer ATR fent la mitjana simple dels primers 14 dies de TR, i finalment s'aplica la fórmula recursiva de Wilder per combinar l'ATR anterior amb la True Range actual i omplir la columna ATR del conjunt de dades.

S'utilitza la següent funció per calcular l'ATR:

```
def calculate_atr_recursive(df, period=14):
    # Calcular el True Range (TR)
    high_low = df['High'] - df['Low']
    high_prev_close = abs(df['High'] - df['Close'].shift(1))
    low_prev_close = abs(df['Low'] - df['Close'].shift(1))
    tr = pd.concat([high_low, high_prev_close, low_prev_close], axis=1).max(axis=1)

    atr_values = [] # Llista on anirem emmagatzemant els valors d'ATR

    # Per a les primeres 'period' files no es pot calcular ATR de manera recursiva
    for i in range(len(tr)):
        if i < period:
            atr_values.append(np.nan) # Assignem NaN fins que tinguem suficients dades
        elif i == period:
            # Primer valor d'ATR: mitjana simple dels TR dels primers 'period' dies
            initial_atr = tr.iloc[i - period + 1:i + 1].mean()
            atr_values.append(initial_atr)
        else:
            # Calcular l'ATR recursiu
            atr_today = ((atr_values[i - 1] * (period - 1)) + tr.iloc[i]) / period
            atr_values.append(atr_today)

    df['ATR'] = atr_values
    return df
```

Fig. 15. Funció de Python per calcular l'ATR.

En aquesta gràfica que calcula l'ATR veiem evidència de pics de volatilitat durant les caigudes de 2020 i 2022, una moderació el 2023 i un nou repunt cap el 2025.

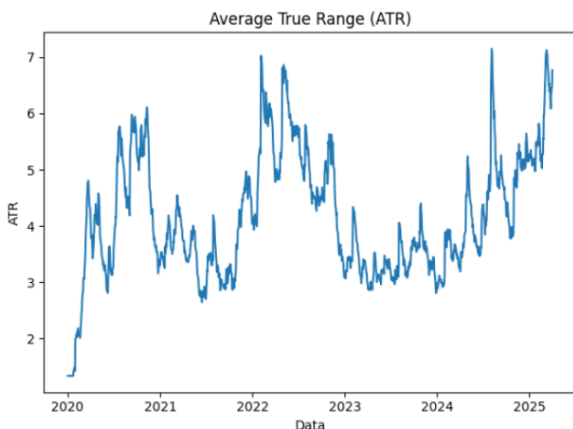


Fig. 16. Gràfica de l'indicador ATR generat per al conjunt de dades d'Amazon.

4.2 Models de Predicció

A continuació, es detallen els diferents models d'aprenentatge automàtic i d'aprenentatge profund implementats per predir el preu de les accions a curt termini (10 dies). Aquests, han generat prediccions per als conjunts de validació i prova i s'han avaluat utilitzant les mètriques explicades anteriorment. Una vegada entrenats els models, s'ha procedit a generar les prediccions del preu de les accions per als deu dies laborals següents, corresponents en aquest cas, als 10 primers dies laborables de febrer. Això es fa per posar a prova la capacitat de cada model per anticipar l'evolució a curt termini dels preus borsaris. Finalment, s'han generat gràfics interactius (*Plotly*) amb les prediccions comparades amb els valors reals, així com gràfics de

predicció futura.

4.2.1 Ridge Regression

Primer de tot, es comença per separar el conjunt de variables d'entrada de la variable de sortida, que en el nostre cas és el preu de tancament (*Close*). Seguidament, es genera la segmentació cronològica del conjunt de dades en 70 % entrenament, 15 % validació i 15 % prova.

Mitjançant un pipeline de *Scikit-Learn* que combina *StandardScaler* per normalitzar la influència de cada variable i *Ridge*, es realitza una cerca exhaustiva del millor factor de regularització α mitjançant validació creuada amb cinc particions seqüencials (*TimeSeriesSplit*). Un cop determinat el millor paràmetre, el model s'entrena sobre el conjunt d'entrenament i s'avalua sobre validació i prova, calculant i desant les mètriques RMSE, MAE i R^2 .

Finalment, es realitza *forecast autoregressiu* per als 10 dies següents. Cada nou dia pren el tancament predicció com a *Close*, es recalculen els indicadors i s'alimenta novament el model per obtenir la següent predicció.

4.2.2 Random Forest

Abans d'entrenar el model, es creen retards (*lags*) de la variable *Close* per als dies 1, 2, 3, 5 i 10, se separa el conjunt de dades en entrenament, validació i prova i s'aplica una tècnica d'augment de dades (*jittering*) que genera diverses còpies amb soroll gaussià lleuger per reforçar la variabilitat de les característiques.

L'arquitectura es basa en una cerca aleatòria de l'espai d'hiperparàmetres de *RandomForestRegressor* amb *RandomizedSearchCV* i validació creuada de sèries temporals per afinar paràmetres com el nombre d'arbres, la profunditat i el mostreig de variables. Per obtenir millors resultats, una vegada entrenat el model de *Random Forest* i avaluat sobre prova, s'obté la importància de les característiques del model i se seleccionen les que tenen un pes major a 0.01. Llavors, es construeix un segon model reduït només amb aquestes variables i es torna a avaluar.

Com podem observar que amb aquest model reduït obtenim millors mètriques de predicció respecte al model complet amb totes les variables, es sobre aquest model optimitzat que es crea finalment la implementació d'un esquema d'autoregressió. Per als propers 10 dies es prenen només els *lags* seleccionats, acumulant cada nova predicció com a valor anterior per alimentar la següent i així successivament fins a completar el *forecast* de deu dies.

4.2.3 SVR

Es comença un altre cop per separar el conjunt de dades en 70 % per entrenament, 15 % per validació i 15 % per prova, separant-les seqüencialment sense barrejar l'ordre temporal. Per garantir que tant les variables d'entrada (*Open*, *High*, *Low*, *Volume* més els indicadors tècnics) com la de sortida (*Close*) tinguin mitjana zero i desviació estàndard un, s'aplica *StandardScaler*.

Es crea un SVR(*kernel*='rbf') i es defineix una quadrícula de valors per a *C*, *gamma* i *epsilon*. Amb *TimeSeriesSplit*(*n_splits*=5) dins d'un *GridSearchCV*, el model prova totes les combinacions respectant la cronologia i busca minimitzar l'error quadràtic negatiu (*neg_mean_squared_error*). D'aquesta manera obtenim la combinació triple (*C*,

γ , ϵ) que millor generalitza. Amb els hiperparàmetres òptims es reentrena un SVR definitiu sobre el conjunt d'entrenament complet, es prediu sobre validació i prova, i se'n calculen les mètriques d'avaluació.

Per acabar, es genera *forecast autoregressiu* a 10 dies que es basa en la capacitat del model d'entendre el comportament passat del *Close* mitjançant indicadors tècnics recalculats a cada iteració. Agafa els últims indicadors (*EMA_7*, *EMA_40*, *MACD*, *Signal_Line*, *MACD_Histograma*, *RSI* i *ATR*, *Volume*), els escala, el SVR calcula el tancament pronosticat i després aquest valor s'integra com a nova fila sobre la qual es calculen els indicadors, generant automàticament la següent predicció fins a completar la sèrie de deu dies.

4.2.4 LSTM

Per a la creació d'aquest model, es comença per crear finestres desplaçades de longitud $N_STEPS=30$. Seguidament, es divideix el conjunt en dades d'entrenament, validació i prova preservant l'ordre cronològic i s'escalen les dades utilitzant *MinMaxScaler* per a les X i un segon scaler per a la Y quedant totes les dades en l'interval $[0,1]$.

Per trobar la millor arquitectura, es fa una búsqueda aleatòria d'hiperparàmetres on es proven 20 arquitectures diferents (1-3 capes LSTM, 64-512 unitats, *dropout* 0.1-0.5, lr 1e-5-1e-3). Cada model s'entrena fins a 50 èpoques amb *EarlyStopping* i *ReduceLROnPlateau*, i es conserva el que minimitza la pèrdua de validació.

Es guarda el millor model i es calculen les mètriques MAE, RMSE i R^2 sobre el conjunt de prova i es genera el *forecast autoregressiu* a 10 dies. S'agafa l'última finestra de 30 dies, es prediu el següent dia de tancament, es reescala a valor real i s'afegeix al conjunt de dades recalculant els indicadors corresponents, repetint aquest bucle fins a completar els 10 dies laborables següents.

4.2.5 GRU

Primer de tot, es comença per separar cronològicament el conjunt de dades en entrenament, validació i prova, seguidament es normalitzen X i Y amb *StandardScaler* i es generen seqüències temporals de longitud $look_back = 20$.

Utilitzant *KerasTuner* es fa una cerca aleatòria de 20 bucles dels hiperparàmetres: filtres i mida del nucli de la capa *Conv1D*, unitats GRU bidireccionals, taxa de *dropout*, regularització $L2$, unitats de la capa densa i *learning rate*. Cada prova s'entrena fins a 50 èpoques amb *EarlyStopping* sobre validació. Un cop trobat el millor model, es fa un *fine-tuning* entrenant-lo fins a 100 èpoques sobre el conjunt d'entrenament i validació i es calculen MAE, RMSE i R^2 .

Finalment, per fer el *forecast autoregressiu* de 10 dies, es pren l'última finestra de 20 dies escalada, es prediu el següent tancament, s'afegeix al conjunt de dades com a nova fila, se'n recalculen els indicadors, es torna a extreure la finestra més recent per a la següent iteració i això es repeteix fins a completar els deu dies.

4.2.6 Híbrid: LSTM + XGBoost

La idea d'aquest model híbrid és que l'*XGBoost* aprengui just allò que la LSTM no ha sabut encertar: els residus. Un cop entrenada la LSTM tal com s'ha explicat anteriorment, es calculen els residus (la diferència entre els valors reals i

les prediccions de la LSTM) sobre les dades d'entrenament i s'utilitza un regressor *XGBoost* per ajustar aquests residus, de manera que aprengui els patrons de predicció que no han estat captats per a la LSTM. Com a últim pas, la predicció final del model s'obté com la suma de la sortida de la LSTM i la sortida de *XGBoost*.

Per al *forecast autoregressiu* híbrid a 10 dies fem exactament el mateix bucle que per a la LSTM però substituint la suma "*LSTM + XGBoost sobre residus*" en cada pas.

5 RESULTATS

Per a tots els models, s'han calculat les tres mètriques d'error diferents: MSE (Error Quadràtic Mitjà), MAE (Error Absolut Mitjà) i R^2 (Coeficient de determinació). Els resultats d'avaluació en prova de tots els models per a cadascun dels conjunts de dades es poden consultar a les taules de l'Apèndix A1.

A continuació, es mostren tres gràfiques que recullen la mitjana del MAE, RMSE i R^2 per a cada model, calculada sobre tots els conjunts de dades. Aquestes figures permeten comparar fàcilment el rendiment mitjà dels models, destacant el model híbrid LSTM+XGBoost i el SVR com els més precisos, i el *Random Forest* com el menys eficient en general.

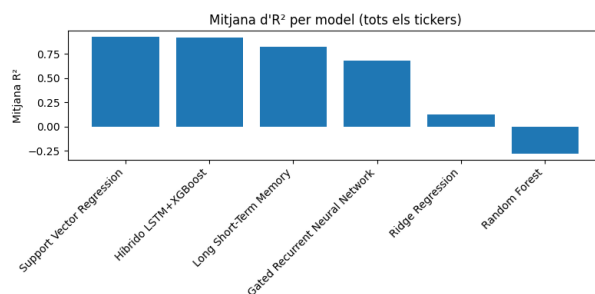


Fig. 17. Comparació de la mitjana de R^2 per als sis models de predicció.

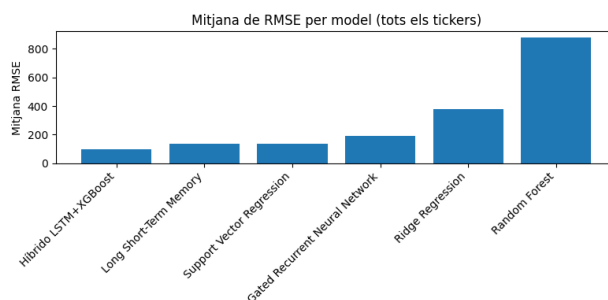


Fig. 18. Comparació de la mitjana de RMSE per als sis models de predicció.

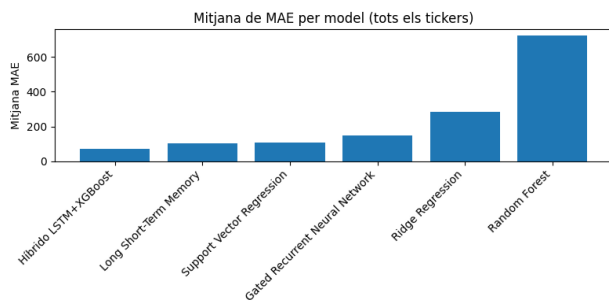


Fig. 19. Comparació de la mitjana de MAE per als sis models de predicció.

Analizant els resultats d'avaluació en prova podem extreure un seguit de conclusions. Si ens fixem en la mètrica R^2 , que mesura la capacitat explicativa del model, podem observar que per a Amazon, S&P 500 i P&G, l'SVR és el model que obté els millors resultats amb valors de R^2 molt propers a 1 i errors baixos. Aquestes bones puntuacions també es reflecteixen en els seus RMSE, amb valors com 2.23 per Amazon o 0.47 per P&G, que indiquen una capacitat notable de predicció en actius amb certa estabilitat o menor volatilitat.

Tanmateix, per a Google, Euro Stoxx 50, Indra i l'IBEX 35, lidera el rànquing el model Híbrid LSTM + XGBoost, amb R^2 que van del 0.87 al 0.95 aproximadament. Aquest sobresurt en actius més volàtils, maximitzant l'explicació de la variància (R^2) tot mantenint valors de MAE reduïts i RMSE baixos.

Hi ha un únic cas, pel conjunt de dades de Hang Seng, on la predicció es genera millor utilitzant el model *Random Forest* amb un R^2 de 0.9873. Tot i això, veiem que els models *Random Forest* i *Ridge*, a grans trets, no són recomanables; tots dos presenten R^2 sovint negatives o molt baixes (< 0.75) i valors de RMSE i MAE molt alts (superant els 800 punts en diversos conjunts) que els descarten com a predicció fiable mostrant una forta inestabilitat i poca fiabilitat.

També és interessant veure que els resultats de R^2 del model híbrid, en general, se situen molt a prop o per damunt de la LSTM pura, demostrant que XGBoost aprèn bé els residus. L'LSTM tendeix a funcionar millor que la GRU (R^2 més alts entre un 10-20 % i errors més baixos) en tots els actius provats, probablement per la seva capacitat de retenir memòria a llarg termini.

6 APLICACIÓ WEB

Com s'ha explicat durant tot el treball, un primer objectiu és poder predir el preu de les accions dels 10 dies següents per a cadascun dels conjunts de dades. En concret, s'han entrenat sis models (*Ridge*, *Random Forest*, *SVR*, *LSTM*, *GRU* i un híbrid *LSTM+XGBoost*) sobre vuit sèries històriques (Amazon, Google, Euro Stoxx 50, Hang Seng, IBEX 35, Indra, P&G i S&P 500) i s'ha obtingut la predicció dels deu primers dies laborables de febrer de 2025.

Llavors, per a cada conjunt de dades i per a cada un dels models, s'han predit aquests deu preus i s'ha generat una gràfica on es poden visualitzar. Per poder mostrar això d'una manera interactiva s'ha creat un prototip d'una aplicació web amb *Streamlit*.

L'aplicació arrenca mostrant dues llistes desplegable a la barra lateral, una per seleccionar el conjunt de dades i una altra per escollir el model. Quan l'usuari prem el botó "Mostra prediccions & plots" el codi llegeix amb *pandas* el CSV preprocessat corresponent al conjunt de dades escollit i es construeix la ruta al directori de resultats del model seleccionat (directori penjat a *GitHub*) i dins d'aquesta carpeta s'identifiquen automàticament el fitxer CSV que conté les prediccions a 10 dies i el fitxer *HTML* amb la gràfica interactiva del preu dels 10 dies i la gràfica prova que mostra els valors reals en vers els predits. Amb *st.dataframe()* es mostra la taula de prediccions i amb *st.components.v1.html()* s'afegeixen les gràfiques *HTML* de la comparació entre preus reals i preus predits perquè l'usuari pugui explorar-les. Finalment, l'aplicació també busca i carrega la imatge *PNG* de l'evolució històrica del preu (que hem generat prèviament i col·locat a la carpeta "Gràfiques Preus Històrics") i la mostra amb *st.image()*. Tots els fitxers CSV originals es queden en memòria amb un decorador *@st.cache_data* perquè, si l'usuari canvia de model però manté el mateix conjunt de dades, no cal tornar a carregar-los des del disc. D'aquesta manera la interfície es manté àgil i coherent: l'usuari només escull conjunt de dades i model, prem el botó i veu immediatament la predicció a 10 dies, els gràfics interactius i l'històric de preus. A l'Apèndix A2 hi trobareu les imatges d'un exemple del prototip de l'aplicació web.

7 CONCLUSIONS

En base als resultats obtinguts i al prototip desenvolupat, es poden extreure les següents conclusions:

S'ha pogut comprovar que utilitzar models d'intel·ligència artificial per predir com evolucionaran els preus de les accions a curt termini resulta eficient. Per exemple, el model de suport vectorial (SVR) funciona força bé amb actius com Amazon, P&G i S&P 500, ja que es poden observar els millors valors de R^2 , tots superiors al 0.98 i errors més baixos. Per contra, per a un nombre major de conjunts de dades, com per Google, Euro Stoxx 50, Indra i l'IBEX 35, el model híbrid *LSTM+XGBoost* supera a la resta, combinant la capacitat de la LSTM per manejar seqüències amb la força de XGBoost per corregir errors.

També es pot veure que els models purament seqüencials, com LSTM i GRU, donen bons resultats, tot i que LSTM sol ser una mica millor gràcies a com es gestiona la memòria. Tanmateix, si afegim una segona capa de regressió amb XGBoost per corregir residus, aconseguim una precisió molt millor, sobretot quan els comportaments del mercat són no lineals o canvien bruscament. En canvi, la *Ridge Regression* i el *Random Forest* no són gaire fiables a curt termini, ja que sovint tenen R^2 molt baixos o negatius i errors elevats.

Pel que fa a l'aplicació web que s'ha generat utilitzant *Streamlit*, ha demostrat ser una eina molt útil per veure i comparar models i actius de manera ràpida. La interfície, que és senzilla i combina taules, gràfics *HTML* interactius i imatges històriques, ajuda a veure ràpidament quina pre-

dicció és millor per a cada acció. La càrrega de dades i resultats està optimitzada usant `@st.cache_data`, i la forma en què està dissenyada permet afegir nous models o sèries temporals sense haver d'ensenyar el codi, fent-la fàcil d'expandir amb el temps.

Finalment, per continuar desenvolupant aquest TFG, hi hauria dues futures línies d'investigació que es podrien implementar:

- Crear un bot per provar en temps real els algorismes de *Machine Learning*.
- Integrar anàlisi de sentiments per complementar les prediccions amb informació qualitativa del mercat.

8 GITHUB

Tots els models, els conjunts de dades utilitzats, els resultats (mètriques i gràfiques) i el codi del prototip d'aplicació es poden trobar emmagatzemats al repositori de GitHub:

https://github.com/mmonsocadena/TFG_PredictStock

BIBLIOGRAFIA

- [1] G. Sonkavde, D. S. Dharrao, A. M. Bongale, S. T. Deokate, D. Doreswamy, and S. K. Bhat, "Forecasting Stock Market Prices using Machine Learning and Deep Learning Models: A Systematic Review, Performance Analysis and Discussion of Implications," *International Journal of Financial Studies*, vol. 11, núm. 3, 2023.
- [2] W. Jiang, "Applications of Deep Learning in Stock Market Prediction: Recent Progress," *Expert Systems with Applications*, vol. 184, p. 115537, 2021.
- [3] A. Wong, S. Whang, E. Sagre, N. Sachin, G. Dutra, Y. Lim, G. Hains, Y. Khmelevsky, and F. C. Zhang, "Short-Term Stock Price Forecasting using Exogenous Variables and Machine Learning Algorithms," a *Proc. 2023 IEEE Int. Conf. Interactive Computing and Young Technologies in Asia (ICYTA)*, 2023.
- [4] Y. Huang, L. F. Capretz, i D. Ho, "Machine Learning for Stock Prediction Based on Fundamental Analysis," al *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021, pp. 1–10.
- [5] Z. Hong, "Algorithmic Trading: Predicting Stock Market Trends in Real-Time," *Medium*, Nov. 21, 2024. [Online]. Disponible a: <https://medium.com/@zhonghong9998/algorithmic-trading-predicting-stock-market-trends-in-real-time-ec583f41d69f>.
- [6] "Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions," *ScienceDirect*, 2018. [Online]. Disponible a: <https://www.sciencedirect.com/science/article/abs/pii/S0377221717310652>.
- [7] A. Biswal, "Stock Market Prediction using Machine Learning in 2025," *Simplilearn.com*, gen. 20, 2025. [Online]. Disponible a: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/stock-price-prediction-using-machine-learning>.
- [8] K. Li, "Machine Learning for Stock Price Prediction," *Neptune.ai*, gen. 30, 2025. [Online]. Disponible a: <https://neptune.ai/blog/predicting-stock-prices-using-machine-learning>.
- [9] E. Guru, "Alliberant el potencial de la intel·ligència artificial en la previsió de fluxos d'efectiu," *Emagia.com*, gen. 29, 2025. [Online]. Disponible a: <https://www.emagia.com/ca/blog/artificial-intelligence-in-cash-flow-forecasting/>.
- [10] "Predicting Stock Market Trends using Hybrid SVM Model and LSTM with Sentiment Determination using Natural Language Processing," 2019. [Online]. Disponible a: <https://www.semanticscholar.org/paper/Predicting-Stock-Market-Trends-using-Hybrid-SVM-and/7e8eb93db3ffdc98499a9027f32d6ebb26d3cd69>.
- [11] Sathishkumar, R. Girivarman, S. Parameswaran, i V. Sriram, "Stock Price Prediction using Deep Learning and Sentimental Analysis," *Journal of Emerging Technologies and Innovative Research*, 2020. [Online]. Disponible a: <https://www.jetir.org/view?paper=JETIR2005051>.
- [12] PyFi, "Stock Price Forecasting using Machine Learning in Python, 2023," *PyFi.com*, jul. 19, 2023. [Online]. Disponible a: <https://pyfi.com/blogs/articles/stock-price-forecasting-using-machine-learning-in-python-2023>.
- [13] J. Jay, "Stock Price Prediction with LSTM in Python," *Python In Office*, mar. 13, 2023. [Online]. Disponible a: <https://pythoninoffice.com/stock-price-prediction-with-lstm-in-python/>.
- [14] A. Shaw, "Python Project: Building a Real-Time Stock Market Price Prediction System," *Medium*, nov. 13, 2024. [Online]. Disponible a: <https://medium.com/@abhishekshaw020/python-project-building-a-real-time-stock-market-price-prediction-system-6ce626907342>.
- [15] A. Fadheli, "How to Predict Stock Prices in Python using TensorFlow 2 and Keras," *The Python Code*, gen. 8, 2020. [Online]. Disponible a: <https://thepythoncode.com/article/stock-price-prediction-in-python-using-tensorflow-2-and-keras>.
- [16] M. A. Ciftci, "XGBoost-LSTM for Netflix Stock," *Kaggle*, 2023. [Online]. Disponible a: <https://www.kaggle.com/code/mehmetakifciftci/xgboost-lstm-for-netflix-stock>.
- [17] DeepML1818, "Financial Data Analysis and Machine Learning for Stock Price Prediction using Random Forest," *Medium*. [Online]. Disponible a: <https://medium.com/@deepml1818/financial-data-analysis-and-machine-learning-for-stock-price-prediction-using-random-forest-6f90727a36ed>.
- [18] N. Ajay, "Data Science: Understanding Random Forest Machine Learning Model for Stock Price Prediction," *Medium*. [Online]. Disponible a: <https://medium.com/@nikaljeajay36/data-science-understanding-random-forest-machine-learning-model-for-stock-price-prediction-with-88d1b2f93047>.

-
- E-mail de contacte: 1636465@uab.cat
 - Treball tutoritzat per: Albert Romero Sanchez.
 - Curs 2024/2025.
 - Data d'entrega: 29/06/2025.

APÈNDIX

A1. RESULTATS DETALLATS DELS MODELS DE PRE-DICCIÓ

A continuació, es mostren les taules amb les tres mètriques d’avaluació (MAE, RMSE i R²) calculades per a cada model sobre cadascun dels conjunts de dades analitzats.

AMAZON

MODEL	MAE	RMSE	R2
Long Short-Term Memory	4.5752	5.7316	0.9283
Random Forest	7.2932	15.1142	0.7244
Ridge Regression	10.3103	12.8969	0.5283
Support Vector Regression	1.5696	2.2250	0.9859
Gated Recurrent Neural Network	10.9723	14.0063	0.5735
Híbrid LSTM + XGBoost	3.2784	4.2841	0.9597

Taula. A1.1. Comparativa de MAE, RMSE i R² per als sis models de predicció analitzats en el conjunt de dades d’Amazon.

GOOGLE

MODEL	MAE	RMSE	R2
Long Short-Term Memory	3.3198	4.3153	0.9061
Random Forest	14.0345	20.6791	0.0124
Ridge Regression	8.0266	9.9239	0.3348
Support Vector Regression	2.1135	2.6495	0.9526
Gated Recurrent Neural Network	4.6573	5.9024	0.8257
Híbrid LSTM + XGBoost	2.2545	2.8637	0.9592

Taula. A1.2. Comparativa de MAE, RMSE i R² per als sis models de predicció analitzats en el conjunt de dades de Google

EURO STOXX 50

MODEL	MAE	RMSE	R2
Long Short-Term Memory	51.0807	64.9371	0.7725
Random Forest	367.1521	446.9396	-1.001
Ridge Regression	109.3853	134.9289	-0.2169
Support Vector Regression	34.6572	37.2942	0.9061

Gated Recurrent Neural Network	39.2246	50.9519	0.8598
Híbrid LSTM + XGBoost	25.9500	32.4364	0.9418

Taula. A1.3. Comparativa de MAE, RMSE i R² per als sis models de predicció analitzats en el conjunt de dades d’Euro Stoxx 50.

HANG SENG

MODEL	MAE	RMSE	R2
Long Short-Term Memory	348.2260	482.8291	0.8913
Random Forest	151.2366	204.1381	0.9873
Ridge Regression	813.4717	1204.0293	0.2192
Support Vector Regression	104.5009	161.5884	0.9861
Gated Recurrent Neural Network	491.8284	635.4129	0.8095
Híbrid LSTM + XGBoost	207.8854	312.8780	0.9534

Taula. A1.4. Comparativa de MAE, RMSE i R² per als sis models de predicció analitzats en el conjunt de dades de Hang Seng.

IBEX 35

MODEL	MAE	RMSE	R2
Long Short-Term Memory	112.6017	139.6788	0.8667
Random Forest	836.4639	1105.6703	-0.4710
Ridge Regression	402.3997	468.3994	-0.4765
Support Vector Regression	165.6506	179.8551	0.7823
Gated Recurrent Neural Network	126.8956	149.5815	0.8410
Híbrid LSTM + XGBoost	79.8748	96.5862	0.9286

Taula. A1.5. Comparativa de MAE, RMSE i R² per als sis models de predicció analitzats en el conjunt de dades d’Ibex 35.

INDRA

MODEL	MAE	RMSE	R2
Long Short-Term Memory	234.5538	313.4665	0.7487
Random Forest	3887.6871	4533.0646	-2.7771

Ridge Regression	777.7638	1012.0231	0.6015
Support Vector Regression	554.0426	703.2707	0.8036
Gated Recurrent Neural Network	370.6862	482.7811	0.4029
Híbrid LSTM + XGBoost	171.6423	226.0883	0.8700

Taula. A1.6. Comparativa de MAE, RMSE i R² per als sis models de predicció analitzats en el conjunt de dades d'Indra.

P&G

MODEL	MAE	RMSE	R2
Long Short-Term Memory	2.1416	2.6764	0.6456
Random Forest	3.2545	5.0548	0.6808
Ridge Regression	4.3084	5.2440	0.5554
Support Vector Regression	0.3889	0.4665	0.9879
Gated Recurrent Neural Network	1.7199	2.2272	0.7549
Híbrid LSTM + XGBoost	1.1756	1.5142	0.8876

Taula. A1.7. Comparativa de MAE, RMSE i R² per als sis models de predicció analitzats en el conjunt de dades de P&G.

S&P 500

MODEL	MAE	RMSE	R2
Long Short-Term Memory	71.1080	90.2894	0.8303
Random Forest	505.1915	676.4666	-0.4249
Ridge Regression	153.1395	181.2726	0.5791
Support Vector Regression	14.6333	18.4516	0.9957
Gated Recurrent Neural Network	148.7868	172.7577	0.37052
Híbrid LSTM + XGBoost	73.3193	86.9409	0.8360

Taula. A1.8. Comparativa de MAE, RMSE i R² per als sis models de predicció analitzats en el conjunt de dades de S&P500.

A2. PROTOTIP DE L'APLICACIÓ WEB DE VISUALITZACIÓ DE PREDICCIONS

En aquesta secció trobareu les imatges d'un exemple del prototip de l'app web (creada amb Streamlit) que permet explorar de manera interactiva les prediccions a 10 dies per a cada conjunt de dades i model.

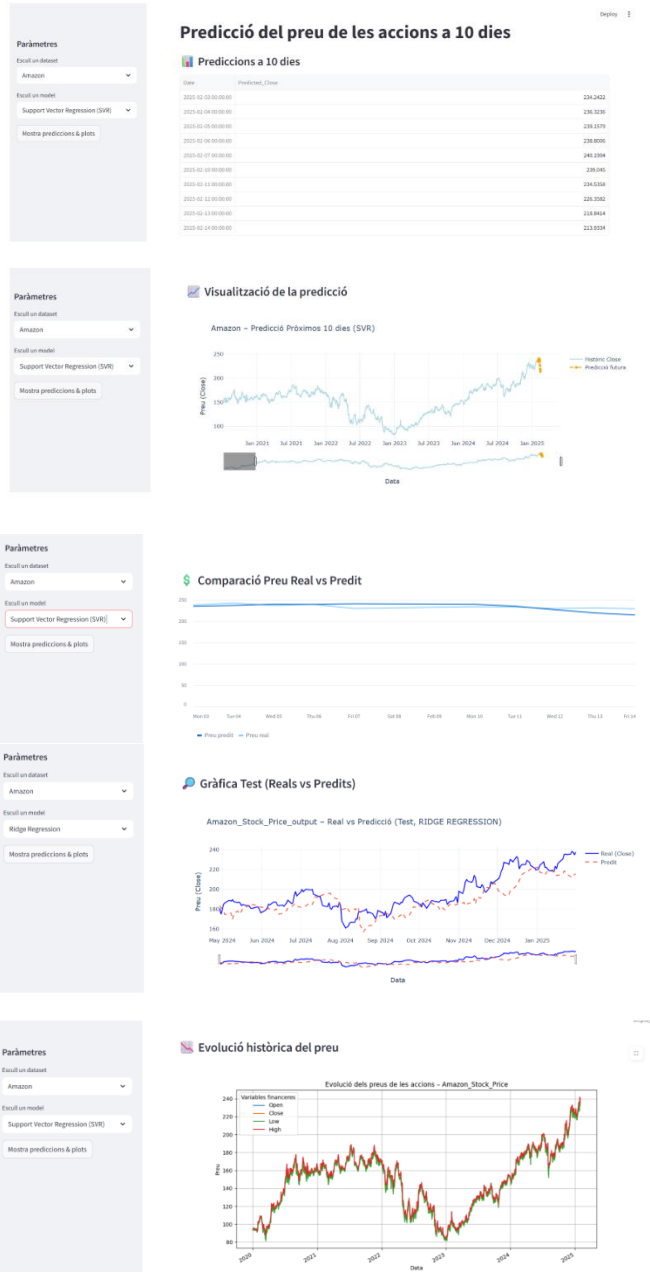


Fig. A2.1. Mostra del prototip de l'aplicació que s'ha creat per visualitzar els preus predits amb els diferents models i per als diferents conjunts de dades. Concretament, en aquest exemple, podem observar els resultats del conjunt de dades d'Amazon obtinguts amb el model SVR.