# Data Science Lab: Vendite E-commerce

Ruben Agazzi 844736
Fabrizio Cominetti 882737
Davide Abete 882299
Tommaso Strada 829351
Alessandro Fasani 837301

28 giugno 2022

**Sommario**

L'obiettivo principale del progetto è quello di valutare e selezionare il miglior modello predittivo relativamente alle stime delle vendite di alcuni settori, precisamente Pesca, Calcio e Casual, di un attività di e-commerce.
Il periodo preso in considerazione va dal giorno 01-01-2014 al giorno 31-12-2021.
Per un attività commerciale presente sulla rete è di fondamentale importanza prevedere le vendite che saranno effettuate nei periodi successivi per ogni settore a disposizione, in modo tale da organizzare disponibilità di magazzino e di spedizione.
I dati sono stati modellati in funzione dell'obiettivo e aggregati secondo diverse granularità, ovvero con frequenza annuale, trimestrale, mensile e settimanale, in modo tale da valutare le performance dei modelli presi in considerazione.
Come sarà possibile verificare nel proseguio del report, la granularità più efficace, relativamente agli obiettivi preposti dal progetto, è risultata quella annuale.
Il progetto è stato effettuato prendendo in considerazione i tre settori con maggiore disponibilità effettiva di osservazioni, ma i modelli utilizzati possono essere applicati anche ad altri settori, per realizzare in questo modo una panoramica completo sull'intero e-commerce.

# Indice



# 1 Introduzione

La realizzazione di questo progetto ha visto come obiettivo principale quello di testare, valutare e decretare il miglior modello predittivo al fine di stimare le vendite in euro realizzate dai vari settori all'interno di un e-commerce.
Prima di tutto, i dati sono stati pre-processati e 'puliti', in modo tale da renderli efficaci e ottimali rispetto allo scopo del progetto. I modelli predittivi, infatti, richiedevano un certo tipo di modellazione dei dati in input per utilizzarli al proprio interno e produrre delle previsioni.
Una volta effettuato il pre-processing dei dati, il focus è passato sul test dei vari modelli, ognuno con le diverse granularità scelte in fase di programmazione.
Il periodo scelto per l'analisi parte dal giorno 01-01-2014 e si protrae fino al giorno 31-12-2021. La scelta di questo periodo non è da considerarsi casuale, in quanto racchiude anche il periodo relativo alla pandemia, periodo che ha visto moltissime persone rivolgersi all'utilizzo di e-commerce, vista l'impossibilità di recarsi in negozi fisici.

## 1.1 Punti Principali

- Vendite: totale - in euro - di guadagno relativamente a ciascun settore, in base alle vendite effettuate in tale data.
- Settori: divisione interna all'e-commerce che identifica le varie categorie di articoli in vendita.
- Previsione: le previsioni effettuate sono caratterizzate da un ruolo primario all'interno del paper.

# 2 Obiettivo

L'obiettivo del progetto è quello di identificare il miglior modello predittivo da fornire ai gestori dell'e-commerce in questione.
Conoscere la previsione relativa agli incassi (in euro) dei vari settori all'interno del sito potrebbe infatti essere utile a diversi scopi all'interno dell'azienda, come ad esempio in fase di programmazione e ordini di materiale, ma anche per avere un'idea di quando applicare o meno determinati sconti e promozioni ai vari settori, con il fine di raggiungere gli obiettivi di vendita prefissati.
Il periodo temporale selezionato è molto lungo, comprende infatti 8 anni di rilevazioni. I dati forniti si caratterizzavano di rilevazioni giornaliere, perciò abbiamo optato per questa scelta temporale in modo da sfruttare a pieno la lunghezza del periodo fornito.
Il progetto può considerarsi rivolto direttamente ai gestori dell'e-commerce e a coloro che al suo interno valutano vendite e ricavi, così come programmazione e ordini.

# 3 Aspetti Metodologici

I modelli selezionati ed utilizzati all'interno del progetto sono i seguenti: ARIMA, TBATS,

PROPHET, XGBOOST.
Di seguito osserviamo i modelli in modo più approfondito da un punto di vista teorico.

## 3.1 ARIMA

In statistica per modello ARIMA (acronimo di AutoRegressive Integrated Moving Average) si intende una particolare tipologia di modelli atti ad indagare serie storiche che presentano caratteristiche particolari. Fa parte della famiglia dei processi lineari non stazionari.
Un modello ARIMA(p,d,q) deriva da un modello ARMA(p,q) a cui sono state applicate le differenze di ordine d per renderlo stazionario. In caso di stagionalità nei dati si parla di modelli SARIMA o ARIMA(p,d,q)(P,D,Q).
Dunque, il modello ARIMA nasce aggiungendo l'integrazione (I) alla combinazione dei modelli autoregressivo (AR) e a media mobile (MA). Il modello ARIMA è composto dalle seguenti componenti:

**P** l'ordine della componente autoregressiva
**D** grado della differenziazione
**Q** ordine della componente a media mobile

Siccome, appunto, un modello ARIMA integra la componente autoregressiva e di media mobile di una serie storica, può essere così definito:

$$(1 - \sum_{i=1}^{p'} \alpha_i L^i) X_t = (1 + \sum_{i=1}^{q} \theta_i L^i)$$

Un modello integrato ARMA di ordine d è un processo stocastico che diventa stazionario dopo essere differenziato d volte.

Abbiamo creato 12 diversi modelli, in modo da predire i valori settimanali, mensili, trimestrali e annuali dei 3 settori di vendita selezionati, ovver calcio, pesca e casual. I parametri dei modelli ARIMA e SARIMA sono trovati minimizzando l'AIC.

I modelli ottenuti e i loro parametri sono i seguenti:

- Annuale pesca: ARIMA model (0,1,0)(0,0,0)
- Trimestrale pesca: ARIMA model (0,2,1)(0,0,0)
- Mensile pesca: SARIMA model (0,1,1)(1,0,0)
- Settimanale pesca: SARIMA model (2,1,2)(1,0,0)
- Annuale calcio: ARIMA model (0,1,0)(0,0,0)
- Trimestrale calcio: SARIMA model (0,0,1)(1,1,0)
- Mensile calcio: SARIMA model (0,0,1)(0,1,0)
- Settimanale calcio: SARIMA model (1,0,0)(1,0,0)
- Annuale casual: ARIMA model (0,1,0)(0,0,0)
- Trimestrale casual: SARIMA model (1,1,0)(1,0,0)
- Mensile casual: SARIMA model (1,1,3)(1,0,0)
- Settimanale casual: ARIMA model (1,0,0)(0,0,0)

## 3.2 TBATS

Il modello TBATS è in grado di considerare e lavorare con stagionalità multiple e complesse.
TBATS è l'acronimo di Trigonometric seasonality, Box-Cox transformation, ARMA errors, Trend and Seasonal components.
In questo modello, ogni stagionalità è modellata su di una rappresentazione trigonometrica, basata sulla serie di Fourier. Uno dei principali vantaggi di questo approccio è che richiede solo due 'seed', indipendentemente dalla lunghezza del periodo. Un altro vantaggio è la possibilità di modellare effetti stagionali di lunghezza non intera. Ad esempio, data una serie di osservazioni giornaliere, è possibile modellare gli anni bisestili con una stagione di lunghezza 365,25.
Il modello TBATS prende in considerazione varie alternative ed è in grado di adattarsi a diversi modelli. Prenderà in considerazione modelli con le seguenti caratteristiche:

- con e senza trasformazione Box-Cox
- con e senza Trend

- con e senza smorzamento del trend
- con e senza processo ARMA(p,q) utilizzato per modellare i residui
- modello non stagionale
- varie quantità di armoniche utilizzate per modellare gli effetti stagionali

Il modello finale sarà poi scelto utilizzando il criterio di informazione di Akaike (AIC).
Nella figura sottostante possiamo osservare due formulazioni matematiche del modello in questione.

**Figura 1:** *[3]*

$$
\begin{aligned}
y_t &= \ell_{t-1} + \phi b_t + s^1_{t-m_1} + \ldots + s^r_{t-m_r} + \eta_t \\
\ell_t &= \ell_{t-1} + \phi b_{t-1} + \alpha \eta_t \\
b_t &= \phi b_{t-1} + \beta \eta_t \\
s^i_t &= \sum_{k=1}^{n_i} s^i_{t,k} + \gamma_i \eta_t \quad (i = 1, \ldots, r) \\
s^{i*}_t &= \sum_{k=1}^{n_i} s^{i*}_{t,k} + \gamma^*_i \eta_t \quad (i = 1, \ldots, r) \\
\eta_t &= \varphi_1 \eta_{t-1} + \ldots + \varphi_p \eta_{t-p} + \varepsilon_t + \psi_1 \varepsilon_{t-1} + \ldots + \psi_q \varepsilon_{t-q}.
\end{aligned}
$$

**Figura 2:** *[5]*

$$
\begin{aligned}
y_t^{(\lambda)} &= l_{t-1} + \phi\, b_{t-1} + \sum_{i=1}^{T} s_{t-m_i}^{(i)} + d_t \\
l_t &= l_{t-1} + \phi b_{t-1} + \alpha d_t \\
b_t &= \phi\, b_{t-1} + \beta\, d_t \\
d_t &= \sum_{i=1}^{p} \varphi_i d_{t-i} + \sum_{i=1}^{q} \theta_i e_{t-i} + e_t
\end{aligned}
$$

Dove:

$y_t^{(\lambda)}$ - time series at moment $t$ (Box-Cox transformed)
$s_t^{(i)}$ - $i$th seasonal component
$l_t$ - local level
$b_t$ - trend with damping
$d_t$ - ARMA($p$,$q$) process for residuals
$e_t$ - Gaussian white noise

## 3.3 PROPHET

Prophet è un modello dedito alla previsione di serie storiche basata su modelli additivi, dove i trend non lineari sono analizzati con diverse stagionalità. Il modello decompone la serie storica in trend, stagionalità ed effetti di festività.
Può essere quindi considerato un modello di regressione non lineare, della forma:
$y(t) = g(t) + s(t) + h(t) + e(t)$
dove:

**g(t)** descrive una tendenza a tratti lineare delle variazioni non periodiche nei dati delle serie temporali (o termine di crescita)
**s(t)** indica i vari modelli stagionali generati dai cambiamenti periodici come la stagionalità giornaliera, settimanale o annuale
**h(t)** rappresenta gli effetti festivi che si possono verificare su cadenze irregolari, ad esempio su un giorno o su un periodo di più giorni
**e(t)** sono termini di errore, ciò che non viene spiegato dal modello

[4][1]
In altri termini l'equazione può essere così scritta:

**Figura 3**

$$
\begin{aligned}
y(t) = {} & \text{piecewise\_trend}(t) + \\
& \text{seasonality}(t) + \\
& \text{holiday\_effects}(t) + \\
& \text{i.i.d. noise}
\end{aligned}
$$

Il modello Prophet prevede due possibili modelli di tendenze per la componente g(t), un modello di crescita saturante e un modello lineare a tratti. La componente stagionale s(t) fornisce adattabilità al modello consentendo periodiche modifiche basate su cambiamenti della stagionalità infragiornaliere, giornaliere, settimanali e annuali. Questa componente per impostazione predefinita, prevede l'utilizzo dell'ordine 10 per la stagionalità annuale e l'ordine 3 per quella

settimanale. La componente h(t) considera eventi prevedibili dell'anno, ad esempio il Venerdì nero, il Superbowl o Natale. Per utilizzare questa funzione, l'utente deve fornire un elenco personalizzato di eventi festivi e l'incorporazione di questa lista nel modello è fatta supponendo che gli effetti delle vacanze siano indipendenti. Il modello viene stimato utilizzando un approccio bayesiano per consentire la selezione automatica dei change points e di altri parametri del modello, se non esplicitamente specificati. Date le sue caratteristiche performa meglio con le serie temporali che hanno forti effetti stagionali e numerose stagioni di dati storici. Il modello è caratterizzato da diversi vantaggi: è preciso e veloce, per questo viene utilizzato ad esempio in molte applicazioni su Facebook per produrre previsioni affidabili per la pianificazione e la definizione degli obiettivi, completamente automatico, in quanto fornisce una previsione ragionevole su dati disordinati senza sforzo manuale, genera previsioni adattabili fornendo molte possibilità agli utenti di modificare e regolare le previsioni al fine di migliorarle, gestisce bene le variazioni stagionali, ed infine, è robusto nei confronti dei valori anomali e resiliente ai dati mancanti.

### 3.4 XGBOOST

L'algoritmo XgBoost è un'implementazione del gradient Boosted Decision Tree utilizzato in larga parte per problemi di classificazione e regressione.
Sebbene sia nativamente adottato per il Machine Learning questo può altresì essere utilizzato per il time series forecasting previa una trasformazione del dataset in un problema di apprendimento supervisionato (dati di input e label). L'XGBRegressor utilizza un numero di Gradient Boosted Tree (riferito al parametro n_estimators nel modello) per predire il valore della variabile dipendente.
Nella previsione di una time series il modello utilizza uno step di lookback (nel codice indicato come shift_giornaliero, shift_mensile, etc.) per prevedere un numero di step nel futuro. La logica alla base di questo shift temporale utilizzato come variabile di input è applicare un fattore di possibile stagionalità delle vendite. Per esempio nell'aggregazione settimanale può essere usata la settimana precedente come variabile di input o la stessa settimana dell'anno prima.
Inoltre, sempre data la sua funzione nativa nel machine learning, non è possibile utilizzare parametri che randomizzino il dataset (come un k-fold) durante la valutazione in quanto si perderebbe la funzione intrinseca della serie temporale. Per ovviare a questo problema si opta per un metodo statico di divisione del dataset (80-20 nel nostro caso).

## 4 Dati

Il dataset utilizzato per il progetto è il dataset "serie-storiche-ecommerce" ed è un file di tipo CSV (Comma Separated Values).
Il file si presentava con un problema relativo alla divisione dell'importo in euro in due differenti colonne, è stata perciò effettuata una correzione per unire le due colonne citate in un unica colonna.
Considerando la correzione effettuata, all'interno del file sono presenti le seguenti colonne:

**data** contenente la data di rilevazione nel seguente formato: DD/MM/YYYY
**totale** importo in euro dell'incasso di uno specifico settore in quel giorno
**settore** testo che identifica il settore dell'ecommerce di riferimento

Per ciascun settore è dunque presente il totale delle vendite (in euro) effettuate in quella data. Le rilevazioni sono dunque giornaliere e divise per settore. Sono pochi i settori che presentano una fetta consistente di rilevazioni, al contrario, per molti settori il numero di osservazioni è limitato.
Le osservazioni mancanti sono state conside-

rate come giorni con incasso nullo, quindi uguale a zero.

**Figura 4:** *Frequenze 'pesca' per anno*

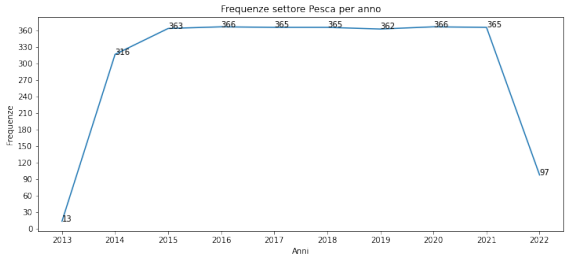



**Figura 5:** *Frequenze 'calcio' per anno*

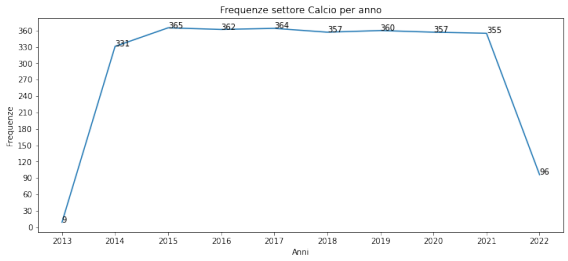
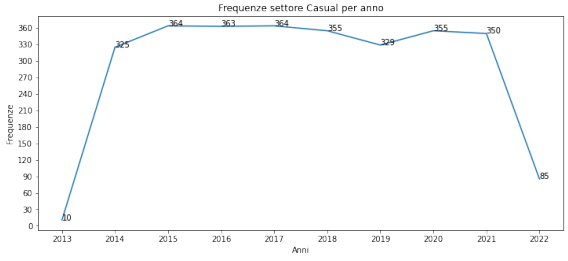



**Figura 6:** *Frequenze 'casual' per anno*

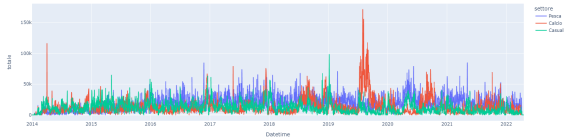



Per questo motivo le analisi successive saranno effettuate considerando i settori con il maggior numero di osservazioni presenti: pesca, calcio e casual.
I dati a disposizione coprono il periodo compreso tra il 2 febbraio 2013 e l'8 aprile 2022. Il file iniziale è composto da un totale di 25262 righe e dalle 3 colonne descritte sopra.

## 4.1 Manipolazione Dati

Durante la fase iniziale di esplorazione dei dati, abbiamo notato la presenza di molti valori mancanti all'interno del dataset, in particolare la maggior parte di questi valori appartenevano all'anno 2013. In questo caso abbiamo deciso di rimuovere completamente l'anno in questione in modo da avere dati continui sugli anni precedenti, evitando di allenare i modelli su dati frammentati e incompleti.
Si è infine scelto di escludere i valori con anno 2022 dal progetto, in quanto incompleti e non utili a raggiungere gli obiettivi prefissati.

**Figura 7:** *Plot serie storiche per settore*




I procedimenti effettuati si possono riassumere dunque nel seguente elenco:

- Unione di colonne erroneamente separate
- Eliminazione osservazioni superflue
- Selezione e divisione dei dataset in base ai settori scelti

## 4.2 Aggreazione dei dati

Lo step successivo è stato dedicato alla creazione dei dataset che saranno utilizzati poi nei vari modelli.
Una volta rimossi i valori precedenti abbiamo raggruppato i dati per i 3 settori di vendita scelti, ovvero pesca, calcio e casual. In seguito i vari dataset già raggruppati per settori sono stati ulteriormente aggregati in base a diversi periodi di tempo, in modo da avere una granularità dei dati più varia. In particolare sono stati creati dataset relativi alle vendite settimanali, mensili, trimestrali e annuali.

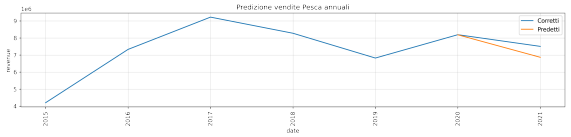
# 5 Analisi per settore

Ora seguirà una breve analisi dei risultati, divisa per settore di vendita, in particolare indicheremo il modello migliore ottenuto per lo specifico settore di vendita. In seguito presenteremo i risultati completi.
Come già specificato, per ciascun settore sono state effettuate previsioni e valutazioni con le quattro diverse aggregazioni temporali.
Per realizzare gli obiettivi prefissati, si è scelto di dividere i dati di train e test seguendo una proporzione 80-20(%).

## 5.1 Analisi Pesca

Nel contesto degli algoritmi usati per il settore di vendita "Pesca", al netto dei raggruppamenti, il modello che ha avuto risultati migliori è 'XGBoost', in particolare applicato su una aggregazione 'annuale' dei dati. Il modello in questione ha avuto un MAPE(Mean Absolute Percentage Error) pari a 4.3%.

**Figura 8:** *XGBoost Pesca Annuale*

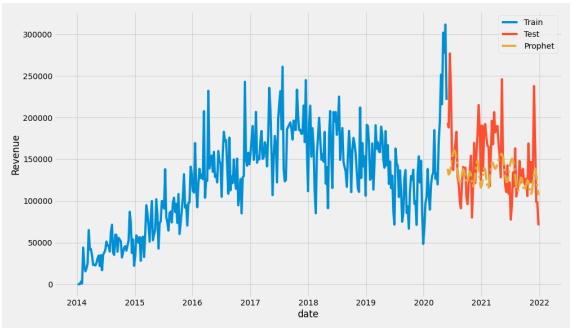



Per quanto riguarda invece i risultati con aggregazione trimestrale, è sempre XGBoost ad ottenere la migliore previsione, così come nel caso di aggregazione settimanale. Il discorso cambia invece con aggregazione mensile, dove è il modello Prophet ad ottenere la migliore precisione. Il settore pesca è quello in cui si ottengono i migliori risultati rispetto agli altri settori.

## 5.2 Analisi Calcio

Per quanto riguarda i dati relativi al settore di vendita "Calcio", il modello migliore tra quelli utilizzati è 'XGBoost' con un Mean Percentage Error pari a 8.33%. Questo modello inoltre ottiene un buon punteggio anche quando applicato ai dati aggregati settimanalmente, con modello Prophet - come osservabile nell'immagine sottostante.

**Figura 9:** *Prophet Calcio Settimanale*

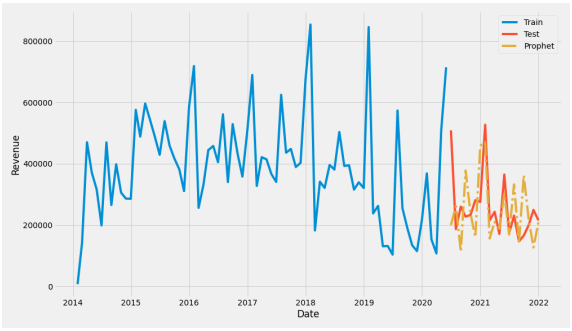



I dati relativi a questo settore non ottengono invece buoni risultati con altri modelli e altri tipi di aggregazioni.

## 5.3 Analisi Casual

Infine, per quanto riguarda il settore di vendite Casual, il modello che ha ottenuto risultati migliori è stato ancora una volta 'XGBoost' applicato ai dati 'annuali'. Il MAPE ottenuto da questo modello è pari a 14.63%.
Nell'immagine proposta di seguito osserviamo invece il modello Prophet, applicato ai dati 'mensili'.

**Figura 10:** *Prophet Casual Mensile*




Anche per quanto riguarda questo settore, valgono le considerazioni fatte per il settore calcio.

# 6 Risultati

Al fine di valutare e selezionare il modello più indicato e preciso relativamente alle finalità del progetto, si è scelto di utilizzare una metrica, 'MAPE', in grado di tener conto anche dell'errore di previsione.
Di seguito il MAPE (Mean Absolute Percentage Error), o errore percentuale medio assoluto, dal punto di vista teorico:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}|\frac{A_t - F_t}{A_t}|$$

dove:

$A_t$ sono i valori reali
$F_t$ sono i valori predetti
**n** rappresenta il numero di osservazioni

La metrica MAPE consiste nella media aritmetica dei rapporti tra il valore assoluto degli errori di previsione e la domanda che si è effettivamente verificata.
Illustriamo di seguito i valori della metrica considerata per ciascuno dei modelli testati.

**Tabella 1:** *MAPE dati annuali*

| | MAPE | | |
|---|---|---|---|
| Modello | Pesca | Calcio | Casual |
| ARIMA | 31.75% | 28.5% | 31.74% |
| TBATS | 28.6% | 50.08% | 28.32% |
| PROPHET | 36.88% | 97.64% | 36.88% |
| XGBOOST | 4.30% | 8.33% | 14.63% |

**Tabella 2:** *MAPE dati trimestrali*

| | MAPE | | |
|---|---|---|---|
| Modello | Pesca | Calcio | Casual |
| ARIMA | 28.67% | 80.68% | 73.66% |
| TBATS | 42.16% | 39.5% | 37.35% |
| PROPHET | 45.87% | 86.19% | 63.67% |
| XGBOOST | 13.08% | 48.01% | 15.87% |

**Tabella 3:** *MAPE dati mensili*

| | MAPE | | |
|---|---|---|---|
| Modello | Pesca | Calcio | Casual |
| ARIMA | 33.60% | 116.59% | 234.71% |
| TBATS | 47.09% | 58.05% | 33.92% |
| PROPHET | 12.86% | 106.61% | 22.88% |
| XGBOOST | 15.87% | 40.19% | 32.49% |

**Tabella 4:** *MAPE dati settimanali*

| | MAPE | | |
|---|---|---|---|
| Modello | Pesca | Calcio | Casual |
| ARIMA | 71.09% | 111.99% | 118.32% |
| TBATS | 40.63% | 159.13% | 41.63% |
| PROPHET | 18.97% | 19.07% | 57.51% |
| XGBOOST | 17.86% | 47.44% | 44.81% |

Alla luce dei risultati visualizzati di sopra, possiamo constatare che il miglior modello predittivo è XGBoost, in grado di garantire un errore previsionale più basso rispetto agli altri modelli, almeno nella maggior parte dei casi analizzati.
Questo vale sia tra i diversi settori che tra le differenti aggregazioni utilizzate.
In conclusione abbiamo notato che il modello generalmente più performante ottenuto è XGBoost applicato ai dati del settore Pesca aggregati annualmente.
D'altro canto il modello che ha avuto la peggiore performance è il modello SARIMA applicato al settore casual con raggruppamento dei dati mensile.

# 7 Conclusioni

L'obiettivo di questo progetto consisteva nell'individuare il miglior modello predittivo tra quelli analizzati, con il fine di renderlo utilizzabile per analizzare le vendite di un e-commerce per i vari settori proposti.
I risultati ottenuti possono essere considerati soddisfacenti, in particolare per i dati con aggregazione annuale.

Il periodo di previsione scelto è molto ampio, ed è possibile aggiornare i risultati nel tempo, con nuove osservazioni.
Ridurre l'intervallo temporale potrebbe garantire un'accuratezza migliore, ma anche tralasciare informazioni importanti.
Potrebbe inoltre essere stimolante estendere l'analisi ad altri settori presenti all'interno dell'e-commerce.
Alcuni ulteriori sviluppi potrebbero consistere nel test di altri algoritmi per la previsione di serie storiche, magari utilizzando anche tecniche di deep learning.

# Bibliografia

[1] Sean J. Taylor Ben Letham. *Prophet: forecasting at scale*. URL: https://research.facebook.com/blog/2017/02/prophet-forecasting-at-scale/.

[2] Jason Brownlee. *How to Use XGBoost for Time Series Forecasting*. URL: https://machinelearningmastery.com/xgboost-for-time-series-forecasting/.

[3] Marco Fattore. *Fundamentals of time series analysis, for the working data scientist*. 2022.

[4] Winston Robson. *The Math of Prophet*. URL: https://medium.com/future-vision/the-math-of-prophet-46864fa9c55a.

[5] Grzegorz Skorupa. *Forecasting Time Series with Multiple Seasonalities using TBATS in Python*. URL: https://medium.com/intive-developers/forecasting-time-series-with-multiple-seasonalities-using-tbats-in-python-398a00ac0e8a.