



UNIVERSITÀ DEGLI STUDI  
DI PERUGIA

DIPARTIMENTO DI INGEGNERIA

Tesi di Laurea in

INGEGNERIA INFORMATICA E ROBOTICA

**Progettazione e sviluppo di un servizio software  
per l'ottimizzazione del processo di chiusura  
contabile aziendale**

Relatore

*FABRIZIO MONTECCHIANI*

Candidato

*GIANLUCA COLETTA*

Anno Accademico 2022/2023



*Alla mia famiglia e alle persone a me care*

# Indice

<b>1</b>	<b>Introduzione</b>	<b>8</b>
1.1	Contributo della tesi . . . . .	9
1.2	Struttura dell'elaborato . . . . .	10
<b>2</b>	<b>Time Series e Forecasting</b>	<b>11</b>
2.1	Caratteristiche delle serie temporali . . . . .	11
2.1.1	Decomposizione . . . . .	11
2.1.2	Stazionarietà . . . . .	13
2.2	Attività di previsione di una serie temporale . . . . .	15
2.3	Confronto tra metodi di previsione statistici, di Machine Learning e di Deep Learning . . . . .	17
2.4	Temporal Fusion Transformer . . . . .	19
2.5	Metriche di valutazione . . . . .	21
<b>3</b>	<b>Tecnologie utilizzate</b>	<b>26</b>
3.1	AutoGluon - TimeSeries . . . . .	26
3.2	Ulteriori librerie Python . . . . .	27
3.2.1	Pandas . . . . .	27
3.2.2	Matplotlib . . . . .	28
3.2.3	Statsmodels . . . . .	28
3.3	Azure Machine Learning Studio . . . . .	29
3.4	Microsoft SQL Server Management Studio . . . . .	30
3.5	Postman . . . . .	31
<b>4</b>	<b>Implementazione del modello</b>	<b>32</b>
4.1	Il set di dati . . . . .	32
4.2	Preprocessing ed Exploratory Data Analysis . . . . .	37
4.3	Model selection . . . . .	44
4.3.1	Risultati model selection . . . . .	50
4.4	Ottimizzazione degli iperparametri . . . . .	51
4.5	Valutazione prestazioni del modello . . . . .	57
4.5.1	Risultati sperimentazione . . . . .	58
<b>5</b>	<b>Ingegnierizzazione della soluzione</b>	<b>62</b>
5.1	Architettura proposta . . . . .	63
5.2	Ottenimento dati . . . . .	65
5.3	Registrazione del modello . . . . .	66
5.4	Creazione servizio . . . . .	67
5.5	Utilizzo del servizio . . . . .	71

---

<b>6</b>	<b>Conclusioni e sviluppi futuri</b>	<b>74</b>
6.1	Conclusioni . . . . .	74
6.2	Limiti e sviluppi futuri . . . . .	75

# Elenco delle figure

1.1	Logo APSIA . . . . .	9
2.1	Esempio di decomposizione di una serie temporale . . . . .	12
2.2	Esempio di serie temporale stazionaria e non stazionaria . . . . .	13
2.3	Obiettivo principale dell'attività di forecasting . . . . .	15
2.4	Previsione media e intervalli di previsione . . . . .	16
2.5	Illustrazione della previsione a più orizzonti . . . . .	19
2.6	Architettura del Temporal Fusion Transformer . . . . .	21
2.7	Riepilogo metriche di valutazione . . . . .	25
3.1	AutoGluon logo . . . . .	27
3.2	Pandas logo . . . . .	27
3.3	Matplotlib logo . . . . .	28
3.4	Statsmodels logo . . . . .	28
3.5	Azure Machine Learning Studio logo . . . . .	29
3.6	Microsoft SQL Server Management Studio logo . . . . .	30
3.7	Postman logo . . . . .	31
4.1	Tabella General Journal Entry . . . . .	33
4.2	Tabella General Journal Account Entry . . . . .	33
4.3	Istogramma relativo al numero di transazioni per ogni filiale da gennaio 2020 ad aprile 2023 . . . . .	37
4.4	Istogramma relativo al numero di transazioni create dall'utente Admin . . . . .	38
4.5	Istogramma relativo al numero di transazioni per ogni singola attività di chiusura contabile . . . . .	39
4.6	Trend del numero di transazioni nel corso del 2020 . . . . .	40
4.7	Trend del numero di transazioni nel corso del 2020 per la filiale Italia . . . . .	41
4.8	Decomposizione additiva della serie temporale . . . . .	41
4.9	Prime cinque righe del dataset contenente le serie temporali . . . . .	42
4.10	Matrice di correlazione . . . . .	43
4.11	Prime cinque righe del dataset finale contenente le serie temporali . . . . .	44
4.12	Porzioni del training set e del test set . . . . .	46
4.13	Cross-validation con numero di finestre di validazione pari a 5 . . . . .	47
4.14	Confronto previsioni del modello con i valori effettivi per la filiale Italia . . . . .	58
4.15	Confronto previsioni del modello con i valori effettivi per la filiale Polonia . . . . .	59
4.16	Confronto previsioni del modello con i valori effettivi per la filiale Francia . . . . .	59
4.17	Confronto previsioni del modello con i valori effettivi per la filiale Gran Bretagna . . . . .	60
4.18	Confronto previsioni del modello con i valori effettivi per la filiale Romania . . . . .	60

---

4.19	Risultati del modello sviluppato per la singola sede Italiana . . . . .	61
5.1	Componenti e funzionamento dell'architettura del servizio . . . . .	64
5.2	Log deploy del servizio web relativo al modello . . . . .	71
5.3	Previsioni numero di transazioni da registrare per la filiale Italia . . . . .	72
5.4	Previsioni numero di transazioni da registrare per la filiale Francia . . . . .	72
5.5	Previsioni numero di transazioni da registrare per la filiale Gran Bretagna . . . . .	73
5.6	Previsioni numero di transazioni da registrare per la filiale Polonia . . . . .	73
5.7	Previsioni numero di transazioni da registrare per la filiale Romania . . . . .	73
6.1	Processo di valutazione e miglioramento del modello . . . . .	76

# Elenco delle tabelle

4.1	Risultati test di stazionarietà . . . . .	42
4.2	Risultati model selection . . . . .	50
4.3	Iperparametri ottimizzati . . . . .	57



# Capitolo 1

## Introduzione

Nel contesto aziendale moderno, la gestione finanziaria riveste un ruolo fondamentale nel garantire la stabilità e la crescita delle organizzazioni. Uno degli aspetti cruciali della gestione finanziaria è rappresentato dal processo di chiusura contabile, un insieme di attività finalizzate a registrare e documentare in modo accurato tutte le transazioni finanziarie di un'azienda al termine di un determinato periodo contabile, che di solito corrisponde a un mese o a un trimestre.

Il processo di chiusura contabile è un complesso insieme di operazioni che richiedono la revisione e l'aggiornamento accurato dei registri contabili, la preparazione meticolosa delle dichiarazioni finanziarie e la creazione di report contabili chiari e precisi. Queste attività sono di vitale importanza per valutare con precisione la performance finanziaria dell'azienda, fornendo informazioni cruciali agli stakeholder e garantendo la conformità normativa e fiscale.

In particolare, per le aziende quotate in borsa, una stretta osservanza dei tempi è imperativa. È essenziale che queste imprese forniscano il bilancio mensile senza alcun ritardo alla fine di ciascun mese. Tale tempestività è cruciale per mantenere la fiducia degli investitori e la trasparenza del mercato finanziario.

Tuttavia, questo processo può essere estremamente laborioso e soggetto a diversi ostacoli. La mole di dati da gestire può essere considerevole, soprattutto in aziende di grandi dimensioni o con un elevato volume di transazioni. Inoltre, la necessità di garantire l'accuratezza e l'integrità delle informazioni finanziarie può richiedere tempi di verifica e revisione prolungati, rallentando il processo nel suo complesso.

Vengono eseguite diverse attività durante la chiusura del bilancio, tra cui la registrazione delle fatture di vendita e degli acquisti, l'ammortamento dei cespiti<sup>1</sup>, la registrazione delle rettifiche e delle allocazioni contabili, la preparazione delle dichiarazioni fiscali e la riconciliazione bancaria.

Un aspetto importante è il principio della partita doppia, che richiede la registrazione di ogni transazione in due parti: l'aspetto economico e l'aspetto finanziario. Questo

---

<sup>1</sup> Rappresenta un bene destinato a essere utilizzato dall'azienda per un periodo di tempo prolungato. A causa della sua durata pluriennale, il costo di acquisto del cespite viene suddiviso su più periodi contabili attraverso il processo di ammortamento contabile.

significa che ogni movimento viene registrato due volte con lo stesso importo, riflettendo sia l'impatto economico, come ad esempio l'insorgenza di un ricavo o un costo, che finanziario, ovvero l'entrata o l'uscita monetaria relativa al ricavo o al costo.

L'idea alla base del progetto è quella di sviluppare un servizio software in grado di ottimizzare il complesso processo di chiusura contabile mensile. Questo servizio mira a fornire una panoramica previsionale del lavoro necessario per completare la chiusura del bilancio entro i tempi previsti in termini di numero di transazioni da registrare a sistema.

Questo approccio offre numerosi vantaggi e benefici tangibili. Innanzitutto, consente una migliore pianificazione delle risorse umane, consentendo di assegnare il personale in modo ottimale a ciascuna attività. Questo previene sovraccarichi e sottoutilizzi, ottimizzando l'efficienza complessiva del processo di chiusura contabile. Inoltre, identifica tempestivamente i punti critici del processo, consentendo l'implementazione di misure correttive per migliorarne l'efficacia. In particolare, permette di individuare le sedi aziendali che possono avere difficoltà nel completare la chiusura mensile in tempo utile, fornendo così un'opportunità per intervenire e migliorare le performance. Infine, una pianificazione più efficiente delle risorse e una chiusura contabile ottimizzata possono portare a una significativa riduzione dei costi operativi complessivi.

Questo lavoro di tesi è stato realizzato in collaborazione con APSIA Italia S.r.l., un'azienda specializzata nella trasformazione digitale e nell'erogazione di servizi indispensabili per il successo dei progetti cloud.



Figura 1.1: Logo APSIA

## 1.1 Contributo della tesi

Il lavoro tesi ha previsto un'attività di ricerca e sviluppo, svolta in un contesto aziendale, sfruttando gli strumenti e le tecnologie messe a disposizione da APSIA. Il risultato finale dell'attività di tesi consiste in un servizio progettato per generare previsioni sul numero di transazioni da registrare nelle prossime dodici settimane, basandosi sull'ultimo set di dati disponibili nel database aziendale.

Le previsioni, elaborate tramite tecniche di forecasting di serie temporali e machine learning, rappresentano il cuore del servizio sviluppato nell'ambito di questa tesi. La complessità delle previsioni richiede un'approfondita analisi dei dati storici e l'applicazione di algoritmi avanzati per identificare pattern e tendenze che possano guidare le proiezioni future. Grazie alla potenza computazionale offerta da Azure Machine Lear-

ning Studio, è stato possibile implementare e ottimizzare tali algoritmi, garantendo la precisione e l'affidabilità delle previsioni fornite.

L'architettura basata su container, integrata nell'infrastruttura Azure Container Services, rappresenta un elemento chiave per la distribuzione e la gestione del servizio. I container consentono una maggiore portabilità delle applicazioni, garantendo la stessa esperienza di esecuzione in diversi ambienti e facilitando la scalabilità orizzontale per gestire carichi di lavoro crescenti. Ciò significa che il servizio può essere facilmente distribuito e integrato in qualsiasi ambiente aziendale, sia esso on-premises o nel cloud, garantendo una maggiore flessibilità e adattabilità alle esigenze specifiche dell'organizzazione.

## 1.2 Struttura dell'elaborato

L'elaborato di tesi è strutturato nei seguenti capitoli e argomentazioni:

- **Capitolo 2:** comprende lo stato dell'arte riguardante il forecasting e le serie temporali. Si descriverà l'attività di previsione di una serie temporale illustrando le caratteristiche principali di queste ultime. Il capitolo si concluderà con un confronto tra le varie tecniche disponibili per effettuare previsioni.
- **Capitolo 3:** vengono presentate le tecnologie utilizzate per implementare il modello e il servizio software.
- **Capitolo 4:** fornisce una panoramica dell'implementazione pratica del modello di forecasting per le serie temporali. Partendo dalla raccolta e preparazione dei dati, vengono descritte le fasi chiave, tra cui la selezione delle caratteristiche, la scelta dell'algoritmo di previsione e l'ottimizzazione dei parametri.
- **Capitolo 5:** vengono mostrate le tecniche usate durante la realizzazione del servizio software, mostrando anche alcuni esempi pratici del suo funzionamento in modo da fornire una visione dettagliata delle metodologie adottate e dei risultati ottenuti.
- **Capitolo 6:** racchiude un riassunto degli obiettivi raggiunti e dei limiti del progetto realizzato, delineando chiaramente i risultati ottenuti e le sfide affrontate durante il processo di sviluppo. Inoltre, saranno esaminati i potenziali sviluppi futuri per ottimizzare e migliorare ulteriormente il servizio, offrendo così spunti per ulteriori ricerche e progressi.

## Capitolo 2

# Time Series e Forecasting

Uno degli obiettivi principali di questo lavoro di tesi consiste nell'approfondire la complessa tematica legata alla predizione di serie temporali. In particolare, una serie temporale si definisce come un insieme di osservazioni ordinate rispetto al tempo, ed esprime la dinamica di un certo fenomeno nel tempo. Le serie temporali vengono studiate sia per interpretare un fenomeno, individuando componenti di trend, di stagionalità e dei residui, sia per prevedere il suo andamento futuro<sup>2</sup>.

Questo capitolo si propone di esaminare in dettaglio l'attività di forecasting di una serie temporale, illustrando le fasi fondamentali coinvolte in questo processo. Inoltre, verranno esaminati i principali approcci utilizzati per effettuare previsioni, evidenziandone le differenze e le peculiarità.

Nello specifico, verranno esaminate diverse famiglie di algoritmi, quelli basati su metodi statistici, quelli basati sul Machine Learning, e quelli basati su Deep Learning, focalizzando l'attenzione sul modello Temporal Fusion Transformer, scelto durante la fase di Model Selection.

## 2.1 Caratteristiche delle serie temporali

In questa sezione vengono descritte alcune caratteristiche relative alle serie temporali, concentrando l'attenzione sulla decomposizione (trend, stagionalità e residui) e sulla stazionarietà di una serie storica.

### 2.1.1 Decomposizione

La decomposizione delle serie temporali è un'operazione statistica che decompone una serie temporale in diverse componenti, ciascuno dei quali rappresenta una delle categorie di pattern sottostanti<sup>3</sup>.

Una serie temporale solitamente viene decomposta in tre differenti componenti [1]:

---

<sup>2</sup> [https://it.wikipedia.org/wiki/Serie\\_storica](https://it.wikipedia.org/wiki/Serie_storica)

<sup>3</sup> [https://en.wikipedia.org/wiki/Decomposition\\_of\\_time\\_series](https://en.wikipedia.org/wiki/Decomposition_of_time_series)

- **Trend:** rappresenta la direzione generale dei dati nel tempo, evidenziando se la serie temporale mostra una crescita, una decrescita o nessuna tendenza particolare nel lungo periodo.
- **Stagionalità:** si riferisce ai pattern ciclici o periodici che si ripetono a intervalli regolari nel corso del tempo. Questi pattern possono essere associati a cicli annuali, trimestrali, mensili, settimanali o giornalieri, a seconda della natura dei dati e del fenomeno in esame.
- **Residui:** rappresentano le fluttuazioni casuali o non sistematiche nella serie temporale che non possono essere attribuite al trend o alla stagionalità. Queste oscillazioni possono essere dovute a fattori casuali, errori di misurazione o altre influenze imprevedibili.

Esistono due tipologie differenti di decomposizione:

- **Additiva:**  $y_t = S_t + T_t + R_t$
- **Moltiplicativa:**  $y_t = S_t \times T_t \times R_t$

Dove  $y_t$  corrisponde ai dati,  $S_t$  è la componente stagionale,  $T_t$  è la componente trend ed  $R_t$  è la componente residua, tutto al periodo  $t$ .

Nella figura 2.1 viene mostrato un esempio di decomposizione additiva di una serie temporale.

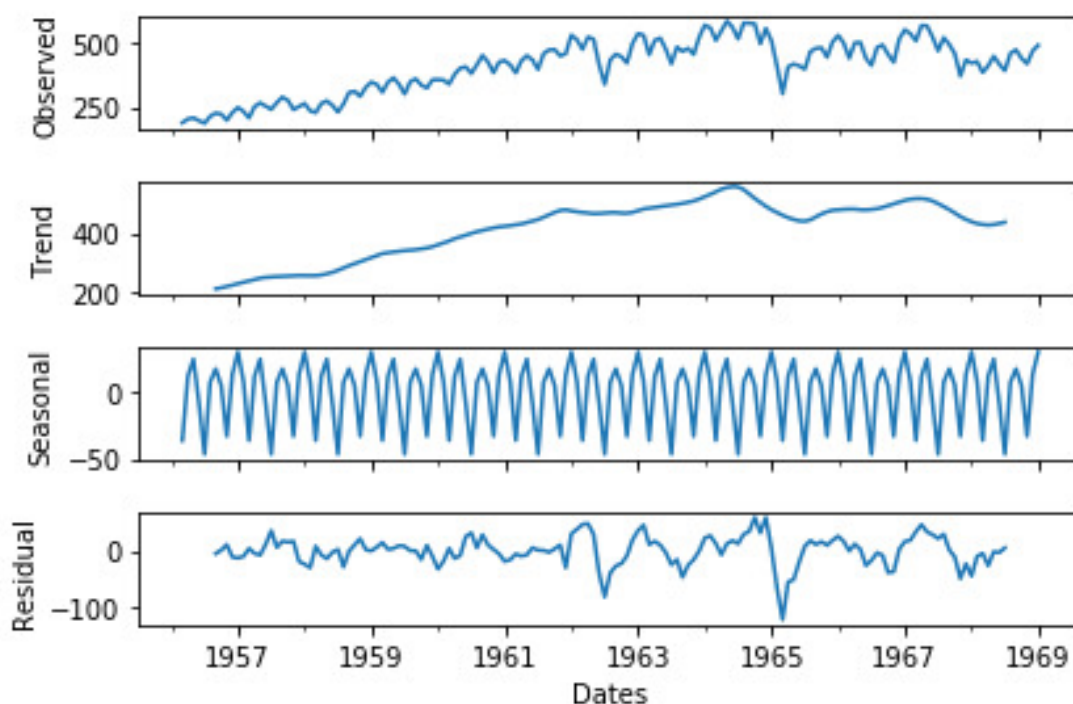


Figura 2.1: Esempio di decomposizione di una serie temporale

### 2.1.2 Stazionarietà

Una serie temporale stazionaria è caratterizzata dal fatto che le sue proprietà statistiche, come media e varianza, rimangono costanti e non dipendono dal momento in cui viene osservata. Viceversa, le serie temporali con trend o stagionalità non sono stazionarie poiché il trend e la stagionalità influenzano il valore della serie in momenti diversi. D'altra parte una serie temporale con rumore bianco è stazionaria, poiché non importa quando viene osservata, avendo lo stesso aspetto in qualsiasi momento.

Quindi, per prevedere valori futuri con una maggior precisione ed efficacia, è preferibile che la serie sia stazionaria, proprio perché mantiene proprietà coerenti indipendentemente dal tempo. Nella figura 2.2 vengono mostrati due esempi di serie temporali, una stazionaria e una non-stazionaria.

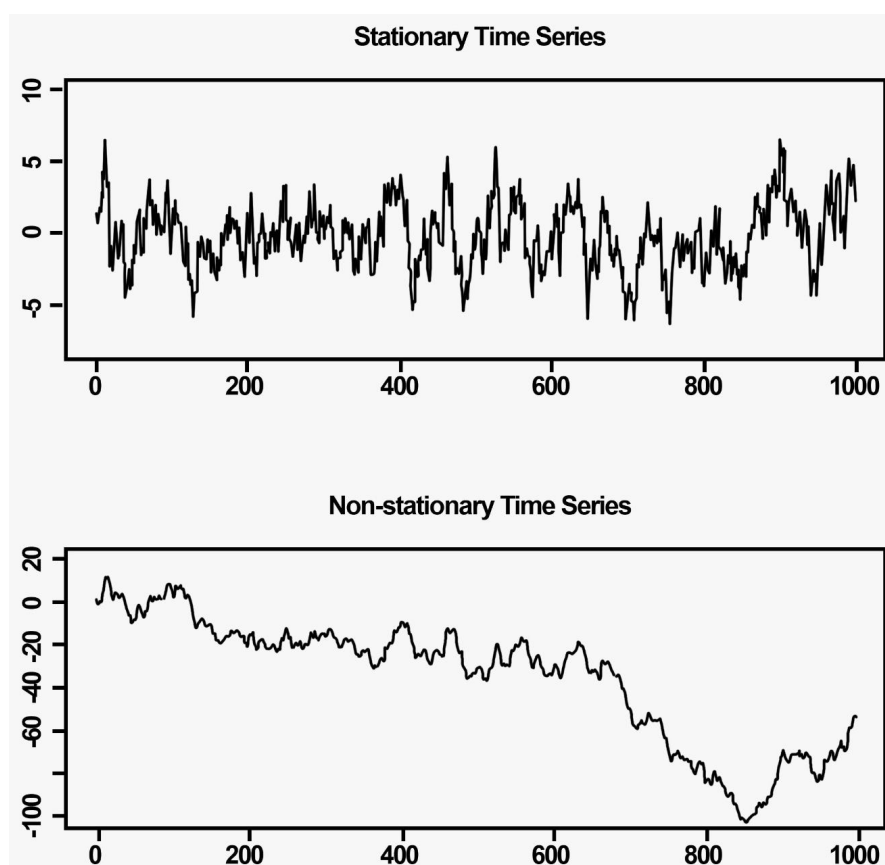


Figura 2.2: Esempio di serie temporale stazionaria e non stazionaria

Per determinare se una serie temporale è stazionaria, è possibile utilizzare diversi metodi di verifica; in questo contesto, è stato adottato l'Augmented Dickey-Fuller (ADF) Test [2], che fa parte di una categoria di test chiamata *Unit Root Test*. Questo test considera due possibili ipotesi in base alla presenza o assenza di una radice unitaria.

L'ipotesi nulla afferma che la serie temporale possiede una radice unitaria, implicando che la serie è non stazionaria. Viceversa, se l'ipotesi nulla viene respinta, si può concludere che la serie è stazionaria. Una radice unitaria è una caratteristica di alcuni processi

stocastici che può causare problemi nell'inferenza statistica che coinvolge modelli di serie temporali. Un processo stocastico lineare ha una radice unitaria se 1 è una radice dell'equazione caratteristica del processo<sup>4</sup>.

In termini tecnici [3], data la seguente equazione matematica:

$$Y_t = \alpha Y_{t-1} + \beta X_e + \epsilon$$

dove,  $Y_t$  è il valore della serie temporale al tempo 't' e  $X_e$  è una variabile esogena (una variabile esplicativa separata), esiste una radice unitaria nella serie temporale se  $\alpha$  assume un valore uguale ad 1.

Dunque la presenza di una radice unitaria renderà la serie non stazionaria. Inoltre, il numero di radici unitarie contenute nella serie corrisponde al numero di operazioni di differenziazione necessarie per rendere la serie stazionaria.

Similmente, il test Dickey-Fuller verifica se, nella seguente equazione, il coefficiente  $\alpha$  è uguale a 1:

$$y_t = c + \beta t + \alpha y_{t-1} + \phi \Delta Y_{t-1} + e_t$$

dove,

- $y_{t-1}$  è il lag 1 della serie temporale;
- $\Delta Y_{t-1}$  è la prima differenza della serie temporale al tempo (t-1).

Il termine lag si riferisce al ritardo o al periodo di tempo che intercorre tra due osservazioni consecutive della serie temporale. In altre parole, rappresenta il numero di periodi temporali passati o futuri rispetto all'istante attuale. La versione aumentata del Dickey-Fuller test, utilizzata in questo progetto, estende l'equazione precedente incorporando un processo regressivo di ordine superiore nel modello.

$$y_t = c + \beta t + \alpha y_{t-1} + \phi_1 \Delta Y_{t-1} + \phi_2 \Delta Y_{t-2} + \dots + \phi_p \Delta Y_{t-p} + e_t$$

Si può notare come l'equazione sia rimasta sostanzialmente invariata, tranne per l'aggiunta dei termini di differenziazione, rendendo il test più completo. Allo stesso modo, l'ipotesi nulla è identica a quella del Dickey-Fuller test. Pertanto, per valutare la veridicità dell'ipotesi nulla (ovvero la presenza di una radice unitaria), è necessario osservare il *p-value*:

- Se il p-value < 0.05 (livello di significatività), l'ipotesi nulla verrà rifiutata;
- Altrimenti, verrà accettata, indicando che la serie non è stazionaria.

<sup>4</sup> [https://en.wikipedia.org/wiki/Unit\\_root](https://en.wikipedia.org/wiki/Unit_root)

## 2.2 Attività di previsione di una serie temporale

La previsione di una serie temporale rappresenta un'attività statistica di fondamentale importanza in molteplici applicazioni, poiché fornisce uno strumento utile per prendere decisioni strategiche e operative. Attraverso l'analisi delle tendenze passate e l'uso di modelli statistici, il forecasting aiuta le aziende a pianificare le attività produttive, i piani di trasporto e la gestione del personale. Inoltre, costituisce una guida alla pianificazione strategica a lungo termine, consentendo alle aziende di adattarsi alle mutevoli condizioni di mercato, anticipare le esigenze dei clienti e rimanere competitive nel panorama aziendale in continua evoluzione. L'obiettivo principale del forecasting è quello di prevedere i valori futuri di una serie temporale basandosi sulle osservazioni passate, come è possibile osservare nella figura 2.3.

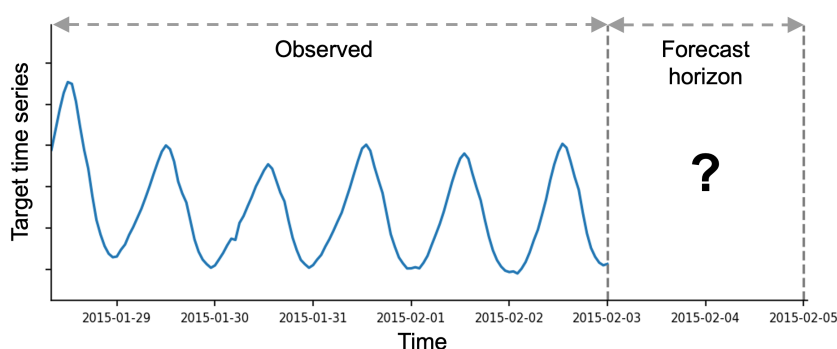


Figura 2.3: Obiettivo principale dell'attività di forecasting

Questa attività comporta solitamente cinque fasi fondamentali [4]:

1. **Definizione del problema:** questa fase spesso rappresenta la parte più complessa delle previsioni. La definizione accurata del problema richiede la comprensione del contesto operativo, di chi richiede le previsioni e di come la funzione di previsione si inserisce all'interno dell'organizzazione che richiede le previsioni.
2. **Raccolta di informazioni:** sono necessari almeno due fonti di informazioni: i dati statistici e l'esperienza accumulata dalle persone coinvolte nella raccolta e nell'utilizzo dei dati. Tuttavia, può essere difficile ottenere dati storici sufficienti per sviluppare un buon modello statistico. In questi casi, si possono utilizzare metodi di previsione basati sul giudizio (*judgmental forecasting*), ovvero sull'esperienza e le conoscenze delle persone coinvolte nel processo decisionale.
3. **Analisi esplorativa:** è un processo mirato di analisi approfondita dei dati storici. L'obiettivo principale è comprendere le caratteristiche intrinseche dei dati e identificare pattern, tendenze e possibili anomalie. Durante questa fase, vengono esaminati grafici dei dati per individuare eventuali modelli coerenti, trend significativi e possibili stagionalità. Inoltre, è possibile valutare eventuali relazioni tra le



variabili disponibili dai dati e identificare eventuali correlazioni o dipendenze che potrebbero influenzare le previsioni.

4. **Scelta e fitting dei modelli:** la scelta del miglior modello dipende dalla disponibilità dei dati storici, dalla forza delle relazioni tra la variabile da prevedere e le variabili esplicative, nonché dalle modalità di utilizzo delle previsioni. Durante la fase di selezione del modello, viene adottato un approccio di *cross-validation*, confrontando le prestazioni di diversi modelli disponibili, sia di tipo statistico che di Deep Learning. L'obiettivo è identificare il modello che massimizza una specifica metrica di valutazione delle prestazioni.
5. **Utilizzo e valutazione di un modello di forecasting:** dopo aver selezionato un modello e ottimizzato i suoi iperparametri, esso viene utilizzato per effettuare le previsioni. Le prestazioni del modello possono essere valutate inizialmente con dei dati di test, ma successivamente sarà necessario valutarlo accuratamente solo dopo aver ottenuto i dati per il periodo di previsione.

Dopo aver completato le varie fasi, è possibile prevedere non solo i valori futuri di una determinata serie temporale, ma anche stabilire gli intervalli di previsione entro i quali è probabile che tali valori si verifichino. Di conseguenza, si possono generare due tipi di previsioni:

- **Previsione media:** rappresenta il valore atteso della serie temporale ad ogni passo temporale dell'orizzonte di previsione.
- **Previsione quantile:** fornisce una stima dei quantili della distribuzione della previsione. Ad esempio, se il quantile 0,1 (noto anche come P10, o 10° percentile) è uguale a  $x$ , significa che il valore della serie temporale è previsto al di sotto di  $x$  il 10% delle volte. Allo stesso modo, il quantile 0,5 (P50) corrisponde alla previsione mediana. I quantili sono utili per comprendere la gamma di risultati possibili: in particolare, in base alla definizione dei quantili, si prevede che la serie temporale sia compresa tra i valori P10 e P90 con una probabilità del 80%.

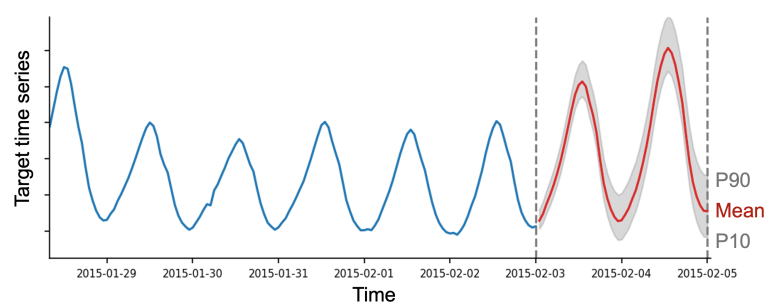


Figura 2.4: Previsione media e intervalli di previsione

## 2.3 Confronto tra metodi di previsione statistici, di Machine Learning e di Deep Learning

L'accuratezza delle previsioni è cruciale in molte applicazioni, e per ottenere risultati ottimali è importante valutare e confrontare diversi approcci di previsione. Verranno esaminati tre categorie principali di metodi di previsione [5]:

- **Statistici:** si basano su analisi delle tendenze e dei pattern nei dati storici per fare previsioni future. Questi metodi spesso utilizzano modelli come ARIMA (Auto Regressive Integrated Moving Average) e modelli esponenziali per modellare le relazioni nei dati e generare previsioni.
- **Machine Learning:** utilizzano algoritmi per apprendere dai dati storici e fare previsioni. Questi algoritmi possono includere alberi decisionali, support vector machine, regressione lineare e altri. La peculiarità di questi approcci risiede nella loro capacità di adattarsi a modelli complessi e non lineare nei dati.
- **Deep Learning:** sono una branca dei metodi di Machine Learning che utilizzano reti neurali profonde per apprendere automaticamente le caratteristiche e le relazioni nei dati. Questi approcci sono noti per le loro capacità di gestire grandi quantità di dati e di modellare relazioni complesse.

In modo più approfondito, il principale punto di forza dei metodi di Machine Learning, se confrontati con quelli statistici, è che invece di descrivere il processo di generazione dei dati sottostante, ad esempio in termini di trend e stagionalità, consentono di identificare e stimare automaticamente le relazioni tra i dati, risultando così più generici [6].

Allo stesso tempo, dato che i metodi di Machine Learning non fanno ipotesi rigide o ne fanno poche riguardo ai dati, la loro performance dipende fortemente dalla disponibilità di dati adeguati, soprattutto quando le serie da prevedere sono non stazionarie e presentano stagionalità e trend. Di conseguenza, per molti anni i metodi di Machine Learning sono stati superati, in termini di prestazioni, da approcci statistici semplici ma robusti. Poiché la disponibilità di dati non è più un fattore limitante e sono stati sviluppati algoritmi più efficaci per l'estrazione di informazioni dai dati, i metodi di Machine Learning e, più recentemente, di Deep Learning, sono diventati alternative promettenti rispetto alle tradizionali tecniche di previsione [7].

Lo stato dell'arte offre numerosi articoli che presentano nuovi metodi di previsione basati su Machine Learning e Deep Learning, i quali hanno dimostrato significativi miglioramenti nell'accuratezza delle previsioni. In particolare, tali approcci sono stati applicati efficacemente in diversi contesti, tra cui l'energia, il mercato azionario e la gestione della domanda in ambito logistico.

Ciononostante, gli studi di ricerca in ambiti diversi dalla previsione aziendale, che spesso coinvolgono serie temporali più brevi e non stazionarie, hanno riportato risultati contrastanti sull'accuratezza delle previsioni. A seconda del contesto del compito di previsione

e dei modelli di Machine Learning coinvolti, si sono riscontrati casi in cui l'utilizzo di tali metodi ha portato a miglioramenti significativi dell'accuratezza [8], mentre in altri casi i risultati sono stati insoddisfacenti [9].

Negli ultimi tempi, i metodi di Deep Learning per le serie temporali stanno diventando sempre più accessibili grazie all'avvento di pacchetti software disponibili pubblicamente per l'uso. Questi pacchetti consentono l'implementazione di approcci di Deep Learning per la previsione delle serie temporali, facilitando il confronto con le metodologie statistiche e di Machine Learning tradizionali. I risultati di uno studio pubblicato nel 2022 [5] hanno messo in luce il potenziale dei metodi di Deep Learning nel contesto delle serie temporali. Questa ricerca ha dimostrato che i metodi sofisticati sono in grado di fornire risultati più accurati rispetto alle metodologie statistiche e di Machine Learning, soprattutto quando viene impiegata una combinazione di più modelli.

Ad ogni modo, nell'era dei Big Data, il tempo di calcolo rimane un fattore critico, specialmente in contesti in cui la velocità dei risultati è altrettanto o addirittura più importante della precisione delle previsioni. In alcune applicazioni, è possibile sfruttare modelli "pre-addestrati" senza necessità di ulteriore training, il che può ridurre significativamente i costi associati all'utilizzo dei metodi di Deep Learning. Tuttavia, in molti casi, l'addestramento del modello è inevitabile e richiede il coinvolgimento di professionisti per sviluppare nuovi modelli o riaddestrare quelli esistenti.

In sintesi, è emerso che i modelli di Deep Learning producono previsioni più accurate rispetto a quelli statistici e di Machine Learning, specialmente per orizzonti di previsione più estesi. Gli ensemble di diversi modelli possono ulteriormente migliorare le prestazioni, sebbene a discapito di costi computazionali più elevati. Allo stesso tempo, un aspetto negativo è rappresentato dal fatto che, nella maggior parte dei casi, il processo di previsione risulta essere una "black-box", con i modelli stimati in modo automatico. Pertanto, sebbene le previsioni derivanti dai modelli di Deep Learning possono essere superiori a quelle dei loro equivalenti statistici e di Machine Learning, non esiste un metodo diretto per gli utenti di comprendere il processo di generazione delle previsioni o per valutare come queste sarebbero cambiate in risposta a variazioni nei fattori di input.

## 2.4 Temporal Fusion Transformer

Tra i numerosi modelli di Deep Learning per fare previsioni su serie temporali spicca il Temporal Fusion Transformer, un'architettura di Deep Neural Network (DNN) basata sull'attenzione per effettuare previsioni a più orizzonti temporali. Questo modello ha dimostrato di ottenere prestazioni eccellenti e introduce nuove forme di interpretabilità nel contesto del forecasting [10]. La rete neurale profonda del Temporal Fusion Transformer utilizza il meccanismo di attenzione per dare maggiore peso a determinate parti dell'input durante il processo di apprendimento. In una DNN tradizionale, ogni neurone in un determinato strato nascosto riceve input da tutti i neuroni dello strato precedente. Ciononostante, nelle reti basate sull'attenzione, vengono introdotti meccanismi che consentono alla rete di concentrarsi su parti specifiche dell'input durante la fase di apprendimento, ignorando o dando meno peso alle altre parti. Il concetto di "attenzione" deriva dall'ispirazione biologica, simile al modo in cui il cervello umano presta attenzione a determinati elementi in un complesso flusso di informazioni. Le previsioni a più orizzonti temporali, a differenza delle previsioni one-step-ahead, forniscono agli utenti una stima non solo del prossimo singolo passo temporale, ma dell'intero percorso futuro per un determinato orizzonte temporale. Questo significa che anziché limitarsi a predire solo il prossimo passo nel tempo, tali previsioni forniscono una visione più ampia e completa delle tendenze e degli eventi futuri lungo un intervallo di tempo specificato. Questo tipo di previsione ha applicazioni significative nella vendita al dettaglio, nell'assistenza sanitaria e nell'economia, dove la capacità di prevedere e pianificare su un orizzonte temporale più lungo può portare a miglioramenti significativi nelle operazioni e nell'efficienza. Inoltre, le applicazioni pratiche di previsione a più orizzonti temporali spesso coinvolgono una vasta gamma di fonti di dati, come mostrato nella figura 2.5, che includono informazioni strutturate, altre serie temporali esogene, e metadati statici, senza alcuna conoscenza preliminare su come interagiscono. L'eterogeneità delle fonti di dati e la limitata comprensione delle loro interazioni rendono estremamente complessa la previsione delle serie temporali a più orizzonti.

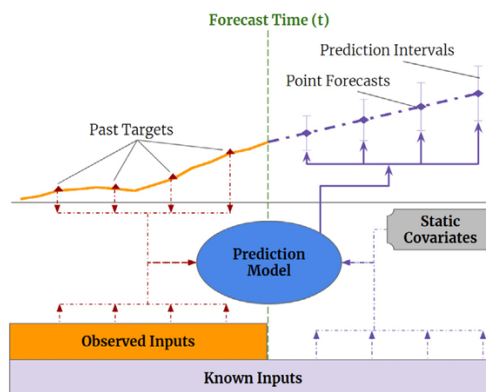


Figura 2.5: Illustrazione della previsione a più orizzonti

La progettazione del Temporal Fusion Transformer include i seguenti componenti principali:

1. **Meccanismi di Gating:** utilizzati per saltare eventuali componenti dell'architettura che non vengono utilizzati, garantendo una profondità e complessità di rete adattive in base alla caratteristiche del dataset e agli scenari di previsione. La relazione precisa tra input esogeni e obiettivi spesso rimane sconosciuta in anticipo, rendendo difficile prevedere quali variabili siano rilevanti. Inoltre, è complesso determinare l'entità dell'elaborazione non lineare richiesta e ci possono essere situazioni in cui modelli più semplici risultano vantaggiosi, ad esempio quando i dataset sono di piccole dimensioni o rumorosi. Con l'obiettivo di conferire al modello la flessibilità di applicare l'elaborazione non lineare solo quando necessario, è stata introdotta la Gated Residual Network (GRN), così fatta:

$$\text{GRN}_\omega(a, c) = \text{LayerNorm}(a + \text{GLU}_\omega(\eta_1)),$$

$$\eta_1 = W_{1,\omega} \eta_2 + b_{1,\omega},$$

$$\eta_2 = \text{ELU}(W_{2,\omega} a + W_{3,w} c + b_{2,\omega})$$

2. **Variable selection networks:** seleziona le variabili di input rilevanti a ogni passo temporale. Sebbene possano essere disponibili molte variabili, la loro rilevanza e il loro contributo specifico all'output sono spesso sconosciuti. TFT è stato progettato per offrire una selezione delle variabili mediante l'uso di reti di selezione delle variabili, sia per le covariate statiche che per quelle dipendenti dal tempo. Oltre a indicare quali variabili sono più significative per il problema di previsione, la selezione delle variabili consente al Temporal Fusion Transformer di eliminare gli input rumorosi non necessari che potrebbero influire negativamente sulle prestazioni. Poiché la maggior parte dei dataset di serie temporali del mondo reale contiene caratteristiche con un contenuto predittivo minore, la selezione delle variabili può notevolmente migliorare le prestazioni del modello, consentendo all'algoritmo di concentrarsi solo sulle variabili più salienti.
3. **Static covariate encoders:** sono progettati per integrare le caratteristiche statiche nella rete, tramite la codifica di vettori di contesto per influenzare le dinamiche temporali. Questi vettori di contesto vengono integrati in diversi punti del decodificatore del Temporal Fusion Transformer, dove le variabili statiche giocano un ruolo cruciale nell'elaborazione. In particolare, si tratta di contesti utilizzati per la selezione delle variabili temporali, l'elaborazione locale delle caratteristiche temporali e l'arricchimento delle caratteristiche temporali con informazioni statiche.
4. **Temporal processing:** per acquisire le relazioni temporali sia a breve che a lungo termine dai dati osservati e noti che si evolvono nel tempo, il modello utilizza

un approccio ibrido. Per l'elaborazione delle informazioni a livello locale, viene impiegato uno strato sequenza-sequenza, in grado di catturare le dipendenze nelle sequenze di dati nel breve termine. Allo stesso tempo, per catturare le dipendenze a lungo termine tra le variabili nel corso del tempo, viene introdotto un innovativo blocco di attenzione multi-head interpretabile. Tale blocco consente al modello di concentrarsi su diverse parti della sequenza temporale, permettendo di catturare relazioni a lungo termine che possono influenzare le previsioni future.

5. **Intervalli di previsione:** il modello utilizza le previsioni quantili, che forniscono una stima della distribuzione dei valori target a ciascun orizzonte temporale. In questo modo, è possibile ottenere un intervallo di probabili valori futuri per ogni punto temporale della previsione.

La figura 2.6 mostra l'architettura ad alto livello del Temporal Fusion Transformer.

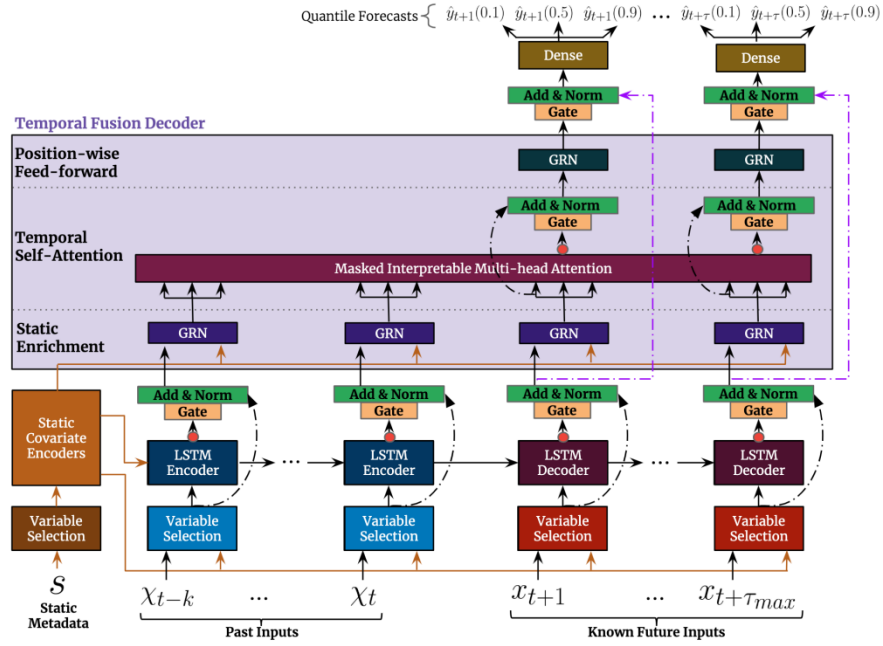


Figura 2.6: Architettura del Temporal Fusion Transformer

## 2.5 Metriche di valutazione

Questa sezione si concentra sull'importanza delle metriche di valutazione nell'ambito della previsione delle serie temporali. Valutare l'accuratezza dei modelli di forecasting è cruciale per comprendere le loro prestazioni e identificare eventuali aree di miglioramento. Le metriche di valutazione forniscono misure quantitative dell'efficacia dei modelli nel fare previsioni e consentono confronti significativi tra diversi approcci di forecasting. In passato sono state proposte molte misure di accuratezza delle previsioni e diversi autori hanno formulato raccomandazioni su quali dovrebbero essere utilizzate quando

si confronta l'accuratezza dei metodi di previsione applicati a dati di serie temporali univariate [11].

La selezione della metrica da adottare può variare in base a diversi fattori, tra cui l'obiettivo della previsione. Ad esempio, se ci focalizziamo sulle previsioni probabilistiche, le metriche adatti includono il Weighted Quantile Loss (WQL) o il Scaled Quantile Loss (SQL). Nello specifico il WQL, conosciuto anche come il weighted pinball loss, calcola la perdita quantile totale divisa per la somma dei valori assoluti delle serie temporali nell'orizzonte di previsione:

$$\text{WQL} = \frac{1}{\sum_{i=1}^N \sum_{t=T+1}^{T+H} |y_{i,t}|} \sum_{i=1}^N \sum_{t=T+1}^{T+H} \sum_q p_q(y_{i,t}, f_{i,t}^q)$$

dove:

- $y_{i,t}$  - valore osservato della serie temporale **i** al tempo **t**
- $f_{i,t}$  - valore predetto della serie temporale **i** al tempo **t**
- $N$  - numero di serie temporali nel set di dati
- $T$  - lunghezza della serie temporale osservata
- $H$  - lunghezza dell'orizzonte di previsione
- $f_{i,t}^q$  - quantile previsto **q** della serie temporale **i** al tempo **t**
- $p_q(y, f)$  - perdita quantile al livello **q**

Queste ultime variabili valgono anche per l'SQL, noto come lo scaled pinball loss, che normalizza la perdita quantile per ciascuna serie temporale in base all'errore stagionale storico di tale serie:

$$\text{SQL} = \frac{1}{N} \frac{1}{H} \sum_{i=1}^N \frac{1}{a_i} \sum_{t=T+1}^{T+H} \sum_q p_q(y_{i,t}, f_{i,t}^q)$$

dove

$$a_i = \frac{1}{T - m} \sum_{t=m+1}^T |y_{i,t} - y_{i,t-m}|$$

Se invece ci interessa una previsione puntuale, possiamo scegliere tra due tipi di metriche: quelle basate sulla stima della mediana e quelle basate sul valore medio.

- **Mediana**

– **Mean Absolute Error:**

$$\text{MAE} = \frac{1}{N} \frac{1}{H} \sum_{i=1}^N \sum_{t=T+1}^{T+H} |y_{i,t} - f_{i,t}|$$

- Mean Absolute Scaled Error:

$$\text{MAPE} = \frac{1}{N} \frac{1}{H} \sum_{i=1}^N \sum_{t=T+1}^{T+H} \frac{|y_{i,t} - f_{i,t}|}{|y_{i,t}|}$$

- Mean Absolute Scaled Error:

$$\text{MASE} = \frac{1}{N} \frac{1}{H} \sum_{i=1}^N \frac{1}{a_i} \sum_{t=T+1}^{T+H} |y_{i,t} - f_{i,t}|$$

dove  $a_i$  è l'errore assoluto stagionale storico.

- Weighted Absolute Percentage Error:

$$\text{WAPE} = \frac{1}{\sum_{i=1}^N \sum_{t=T+1}^{T+H} |y_{i,t}|} \sum_{i=1}^N \sum_{t=T+1}^{T+H} |y_{i,t} - f_{i,t}|$$

- Valore medio:

- Mean Squared Error:

$$\text{MSE} = \frac{1}{N} \frac{1}{H} \sum_{i=1}^N \sum_{t=T+1}^{T+H} (y_{i,t} - f_{i,t})^2$$

- Root Mean Squared Error:

$$\text{RMSE} = \sqrt{\frac{1}{N} \frac{1}{H} \sum_{i=1}^N \sum_{t=T+1}^{T+H} (y_{i,t} - f_{i,t})^2}$$

- Root Mean Squared Scaled Error:

$$\text{RMSSE} = \sqrt{\frac{1}{N} \frac{1}{H} \sum_{i=1}^N \frac{1}{s_i} \sum_{t=T+1}^{T+H} (y_{i,t} - f_{i,t})^2}$$

dove  $s_i$  è l'errore storico stagionale al quadrato.

Sono state analizzate anche le diverse proprietà di ciascuna metrica. Ecco un riepilogo:

- **WQL**: dipendente dalla scala attribuisce più peso agli errori delle serie temporali con valori assoluti più grandi. È equivalente al WAPE quando i livelli del quantile sono impostati su  $[0.5]$ .
- **SQL**: la metrica scalata normalizza l'errore per ciascuna serie temporale sulla scala di quella stessa serie temporale, ma non è definita per serie temporali costanti, poiché non è possibile normalizzare l'errore su una scala costante.



- **MAE**: dipende dalla scala attribuisce più peso agli errori delle serie temporali con valori assoluti più grandi. Tuttavia, non è sensibile agli outlier e preferisce i modelli che stimano con precisione la mediana dei valori.
- **MAPE**: dovrebbe essere utilizzata solo se tutte le serie temporali hanno valori positivi. È indefinita per le serie temporali che contengono valori zero e penalizza le sovrastime più pesantemente rispetto alle sottostime.
- **MASE**: scala l'errore per ciascuna serie temporale in base alla scala di quella serie temporale. È indefinita per le serie temporali costanti e non è sensibile agli outlier.
- **WAPE**: dipende dalla scala, il che significa che le serie temporali con valori assoluti elevati contribuiscono maggiormente alla perdita. Non è sensibile agli outlier e preferisce i modelli che stimano accuratamente la mediana.
- **MSE**: dipende dalla scala, il che significa che le serie temporali con valori assoluti elevati contribuiscono maggiormente alla perdita. Penalizza pesantemente i modelli che non possono adattarsi rapidamente ai cambiamenti repentini nella serie temporale, ed è sensibile agli outlier. Preferisce i modelli che stimano accuratamente la media (valore atteso).
- **RMSE**: stesse caratteristiche del Mean Square Error.
- **RMSSE**: normalizza l'errore per ciascuna serie temporale in base alla scala di quella serie temporale. È indefinita per serie temporali costanti e penalizza pesantemente i modelli che non possono adattarsi rapidamente ai cambiamenti repentini nella serie temporale. È sensibile agli outlier e preferisce i modelli che stimano accuratamente la media.

Infine, un ultimo aspetto da considerare è l'importanza di prevedere con accuratezza serie temporali caratterizzate da valori elevati; in questo contesto è consigliabile utilizzare metriche dipendenti dalla scala come WQL, MAE, RMSE o WAPE. Queste metriche si adattano bene anche a serie temporali sparse che contengono numerosi zeri. Viceversa, se l'obiettivo è trattare equamente tutte le serie temporali nel dataset, si possono prendere in considerazione metriche come SQL, MASE o RMSSE.

Metric	Probabilistic?	Scale-dependent?	Predicts median or mean?
SQL	✓		
WQL	✓	✓	
MAE		✓	median
MASE			median
WAPE		✓	median
MSE		✓	mean
RMSE		✓	mean
RMSSE			mean
MAPE			
SMAPE			

Figura 2.7: Riepilogo metriche di valutazione

## Capitolo 3

# Tecnologie utilizzate

In questo capitolo verranno descritte le tecnologie impiegate nello sviluppo del progetto di tesi. Nei capitoli successivi verrà delineato il processo di ricerca e sviluppo che ha fatto uso di tali tecnologie, accompagnato da una breve esposizione del lavoro svolto complessivamente. Le tecnologie maggiormente impiegate includono:

- AutoGluon - TimeSeries
- Ulteriori librerie Python
- Azure Machine Learning Studio
- Microsoft SQL Server Management

Azure Machine Learning Studio è un servizio cloud reso disponibile da Microsoft sulla sua piattaforma Azure, caratterizzato da una modalità di pagamento del tipo pay as u go, in cui si paga in base alle risorse utilizzate.

### 3.1 AutoGluon - TimeSeries

AutoGluon-TimeSeries è una libreria open-source dedicata alla previsione probabilistica delle serie temporali [12]. Fondata sul principio di Automated Machine Learning (AutoML), questa libreria offre una serie di strumenti e tecniche progettati per automatizzare il processo di sviluppo dei modelli di machine learning. Con un focus specifico sulla previsione delle serie temporali, AutoGluon-TimeSeries si propone di semplificare e velocizzare il processo di creazione di modelli, consentendo agli utenti di generare previsioni accurate utilizzando un numero limitato di linee di codice Python.

Questa libreria è stata sviluppata con un'attenzione particolare alla facilità d'uso e alla robustezza. Basandosi sulla filosofia di design di AutoGluon, AutoGluon-TimeSeries sfrutta una vasta gamma di modelli di previsione diversificati per ottenere un'elevata accuratezza nel minor tempo possibile. AutoGluon-TimeSeries unisce sia modelli statistici tradizionali che approcci di previsione basati sull'apprendimento automatico, insieme a

tecniche di ensemble di modelli multipli. Questa sinergia permette di massimizzare l'utilizzo delle caratteristiche distintive di ciascun modello, portando a un miglioramento complessivo delle prestazioni di previsione.



Figura 3.1: AutoGluon logo

Per utilizzare AutoGluon-TimeSeries sono necessarie solo le due seguenti classi:

- **TimeSeriesDataFrame**: memorizza un insieme di dati composto da una o più serie temporali, specificando la colonna identificativa di ogni serie temporale e la colonna relativa al timestamp.
- **TimeSeriesPredictor**: gestisce il fitting, il tuning e la selezione dei migliori modelli di previsione, oltre alla generazione di nuove previsioni per le serie temporali.

## 3.2 Ulteriori librerie Python

In questa sezione vengono elencate e descritte brevemente le principali librerie Python utilizzate per lo sviluppo del progetto.

### 3.2.1 Pandas

Pandas è una potente libreria open-source scritta per il linguaggio di programmazione Python, utilizzata principalmente per la manipolazione e l'analisi dei dati [13]. Essa fornisce strutture dati flessibili e intuitive, progettate per rendere più semplici operazioni comuni come caricare, manipolare, filtrare e analizzare dati tabulari e serie temporali.



Figura 3.2: Pandas logo

Le principali strutture dati fornite da Pandas sono le Serie e i DataFrame. Le Serie sono array unidimensionali etichettati in cui è possibile immagazzinare dati di qualsiasi tipo (interi, float, stringhe, ecc.) insieme a un indice che rappresenta le etichette delle righe. I DataFrame sono strutture dati bidimensionali simili a tabelle in cui le colonne sono

etichettate e possono contenere dati di diversi tipi. I DataFrame consentono di eseguire operazioni di tipo SQL, come unire, raggruppare e filtrare dati in modo intuitivo.

### 3.2.2 Matplotlib

Matplotlib è una libreria Python ampiamente utilizzata per la creazione di grafici e visualizzazioni di dati [14]. Offre una vasta gamma di funzionalità per la generazione di grafici di alta qualità in diversi formati e stili. Con Matplotlib, gli utenti possono creare grafici 2D e 3D, istogrammi, diagrammi a torta, scatter plot, line plot e molti altri tipi di visualizzazioni.



Figura 3.3: Matplotlib logo

La libreria fornisce un'interfaccia flessibile e potente per la personalizzazione dei grafici, consentendo agli utenti di controllare ogni aspetto dell'aspetto dei loro grafici, tra cui colori, stili di linea, etichette degli assi, titoli e molto altro ancora. Matplotlib è altamente integrato con NumPy e Pandas, rendendo facile la visualizzazione dei dati contenuti in array o DataFrame.

Inoltre, Matplotlib supporta la creazione di grafici interattivi, che possono essere incorporati in applicazioni web o notebook Jupyter per un'esperienza di visualizzazione più dinamica e coinvolgente.

### 3.2.3 Statsmodels

Statsmodels è una libreria Python ampiamente utilizzata per l'analisi statistica e l'adattamento di modelli statistici ai dati [15]. Essa fornisce una vasta gamma di funzionalità per la stima dei parametri, l'analisi dei residui, i test di ipotesi e la visualizzazione dei risultati.



Figura 3.4: Statsmodels logo

Questa libreria è particolarmente utile per eseguire analisi di regressione lineare e non lineare, analisi della varianza (ANOVA), analisi delle serie storiche e molte altre tecniche statistiche. Statsmodels supporta anche modelli per dati a eventi discreti come la regressione logistica e la regressione di Poisson. Offre un'interfaccia intuitiva e facile da usare per definire, adattare e valutare modelli statistici.

Gli utenti possono accedere a una vasta gamma di modelli predefiniti e funzionalità statistiche avanzate, oltre a poter estendere la libreria con modelli personalizzati. Oltre ai modelli, è possibile accedere ad una varietà di strumenti per l'analisi delle serie temporali, inclusi test di stazionarietà, diagnosi dei residui, test di autocorrelazione e visualizzazioni grafiche.

### 3.3 Azure Machine Learning Studio

Azure Machine Learning Studio è un servizio cloud progettato per accelerare e gestire l'intero ciclo di vita dei progetti di apprendimento automatico [16]. Rivolto a professionisti del machine learning, data scientist e tecnici, questo strumento facilita l'esecuzione di flussi di lavoro giornalieri per addestrare e distribuire modelli. Con Azure Machine Learning Studio è possibile creare nuovi modelli direttamente nell'ambiente fornito o utilizzare modelli preesistenti sviluppati su piattaforme open source come PyTorch, TensorFlow o Scikit-Learn.



Figura 3.5: Azure Machine Learning Studio logo

Sono disponibili diverse opzioni di creazione a seconda del tipo di progetto e del livello di esperienza dello sviluppatore. Per massima flessibilità nella progettazione, l'intero lavoro è stato condotto utilizzando un ambiente denominato "notebook", una piattaforma online in cui è possibile salvar file ed eseguire codice.

Il codice sviluppato viene eseguito su risorse cloud, consentendo agli utenti di scegliere dove eseguire il codice ogni volta che lavorano: possono optare per una macchina virtuale dedicata o un cluster di risorse. Tale scelta influisce sui tempi di calcolo, sulle capacità

di gestire volumi di dati e sul costo complessivo dell'operazione. Nella fase di sviluppo, il codice è stato principalmente eseguito su una macchina virtuale.

Nel lavoro sono stati prevalentemente utilizzati due formati: `.ipynb` e `.py`. Il primo consente di scrivere codice Python a blocchi, offrendo la possibilità di eseguire solo porzioni selezionate di codice all'interno di un file. Questo formato è stato particolarmente utile poiché consente di testare singolarmente o in gruppi specifici modelli di previsione delle serie temporali. Tale approccio ha contribuito a ridurre sia i tempi di attesa che i costi computazionali.

Azure Machine Learning Studio offre anche la possibilità di registrare il modello direttamente all'interno della piattaforma e di eseguire il processo di Model Deployment, che consiste nel mettere in produzione i modelli sviluppati. Questa procedura permette di rendere le previsioni del modello accessibili agli utenti, agli sviluppatori o ai sistemi, consentendo loro di prendere decisioni aziendali basate sui dati, di interagire con le proprie applicazioni e di sfruttare appieno il valore dei modelli creati.

### 3.4 Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio è un ambiente integrato per la gestione, lo sviluppo e la configurazione di database SQL Server. Consente agli amministratori di database e agli sviluppatori di accedere e gestire facilmente i database, eseguire query SQL, progettare e modificare schemi di database, monitorare le prestazioni del database e automatizzare le attività di manutenzione.



Figura 3.6: Microsoft SQL Server Management Studio logo

Inoltre, offre un'interfaccia grafica intuitiva insieme a potenti strumenti di scripting e amministrazione per semplificare il lavoro con i database SQL Server [17]. Questo software è stato impiegato per realizzare una piccola base di dati in cui sono state importate le tabelle necessarie per lo sviluppo del modello di previsione delle serie temporali.

### 3.5 Postman

Postman è un applicazione utilizzata principalmente per testare, sviluppare e documentare le API<sup>5</sup>. Questa consente agli sviluppatori di inviare richieste HTTP a un server e visualizzare le risposte, consentendo loro di esaminare e verificare il funzionamento delle API.



Figura 3.7: Postman logo

Nel progetto è stata impiegata per testare il corretto funzionamento del deployment del modello di previsione tramite una richiesta HTTP di tipo POST<sup>6</sup>.

---

<sup>5</sup>Application Programming Interface, interfacce che permettono alle applicazioni di comunicare tra di loro.

<sup>6</sup>Richiesta HTTP per inviare dati a un server web.



## Capitolo 4

# Implementazione del modello

Questo capitolo fornisce una panoramica dell'implementazione pratica del modello di forecasting per le serie temporali. Partendo dalla raccolta e preparazione dei dati, vengono descritte le fasi chiave, tra cui la selezione delle caratteristiche, la scelta dell'algoritmo di previsione e l'ottimizzazione dei parametri. Viene presentato anche il dataset utilizzato nel caso di studio, con un'analisi dettagliata delle sue principali caratteristiche, inclusa la decomposizione della serie temporale e il test di stazionarietà.

Le principali fasi seguite per lo sviluppo del modello di previsione del numero di transazioni registrate nel sistema includono:

1. **Preprocessing ed Exploratory Data Analysis (EDA):** questa fase comprende le operazioni di pulizia del dataset e l'esplorazione dei dati, sia numericamente che visualmente.
2. **Model selection:** confronto tra diversi modelli di forecasting per individuare quello con le migliori performance.
3. **Hyperparameter optimization:** vengono regolati gli iperparametri del modello selezionato per migliorarne le prestazioni.
4. **Model evaluation:** una volta ottimizzati gli iperparametri, il modello viene addestrato sui dati di training e valutato sui dati di test per valutarne l'efficacia.

### 4.1 Il set di dati

Il dataset utilizzato come caso di studio per l'implementazione del modello di forecasting comprende le transazioni registrate al sistema durante la chiusura contabile mensile di una grande azienda che collabora con APSIA, che per questioni di riservatezza sarà identificata con il nome fittizio di Contoso Corporation. Le transazioni registrate includono una vasta gamma di attività finanziarie e contabili, tra cui acquisti, vendite, pagamenti, e altre operazioni rilevanti per l'attività aziendale. Questi dati sono fondamentali

per monitorare e analizzare le performance finanziarie dell'azienda nel tempo nonché per guidare le decisioni operative e strategiche.

L'azienda oggetto di questo studio ha sedi distribuite in varie regioni del mondo. Tuttavia, per il caso di studio in questione, ci si è concentrati su cinque filiali situate in Europa. È importante notare che il lavoro e il progetto possono essere estesi a tutte le altre sedi aziendali. Le sedi europee coinvolte in questo studio sono localizzate nei seguenti paesi: Italia, Inghilterra, Francia, Polonia e Romania.

Le transazioni vengono registrate dagli utenti tramite Microsoft Dynamics 365, software che fornisce strumenti e funzionalità per migliorare la produttività, ottimizzare i processi aziendali e analizzare i dati, il tutto attraverso una piattaforma centralizzata e scalabile basata su cloud.

Le transazioni vengono successivamente archiviate all'interno di un database Microsoft SQL Server, dove sono principalmente coinvolte due tabelle: la GeneralJournalEntry e la GeneralJournalAccountEntry. Di seguito sono riportate le due tabelle appena menzionate.

AccountingDate	Acknowledgement	DocumentDate	DocumentNumber	FiscalCalendarYear	JournalCategory	JournalNumber	Ledger	LedgerEntryJournal	Le...	Le...	PostingLayer	SubledgerVoucher	SubledgerVoucher...	createdDate	createdBy
1/15/2015	1/15/2015			Fiscal 2015	Payment	000001	USMF	5637144576			Current	IV-8002	usmf	1/15/2015 5:08:56 PM	SIMON
1/14/2015	1/14/2015			Fiscal 2015	Purchase order	000001	DEMIF	5637149076			Current	IV-8002	demif	1/16/2015 6:59:53 AM	SUSAN
1/21/2015	1/21/2015	1/21/2015	1000001	Fiscal 2015	Inventory	000001	USRT	5637140908			Current	IV-8002	usrt	1/22/2016 6:06:04 PM	DANIEL
1/15/2015	1/15/2015			Fiscal 2015	Project	000001	USPI	5637182092			Current	IV-8002	uspi	1/15/2015 1:00:51 PM	SUSAN
1/14/2016	1/14/2016	10/1/2016	1000001	Fiscal 2016	Purchase order	000001	FRRT	5637182092			Current	IV-8002	frst	1/16/2017 6:59:06 PM	JUNE
1/22/2016	1/22/2016	1/22/2016		Fiscal 2016	Project	000001	FRSI	5637206852			Current	IV-8002	frsi	2/3/2017 10:21:27 PM	Emmah
1/8/2016	1/8/2016			Fiscal 2016	Project	000001	USPI	5637145350			Current	IV-8002	uspi	1/12/2016 6:41:42 PM	JUNE
1/15/2015	1/15/2015	1/15/2015	SPK-000001	Fiscal 2015	Sales order	000002	USMF	5637145350			Current	IV-8002	usmf	1/15/2015 5:25:43 PM	SIMON
1/15/2015	1/15/2015	1/15/2015	SPK-000001	Fiscal 2015	Sales order	000002	DEMIF	5637145350			Current	IV-8002	demif	1/16/2015 7:16:34 PM	SUSAN
1/21/2015	1/21/2015	1/21/2015	1000002	Fiscal 2015	Purchase order	000002	USRT	5637145350			Current	IV-8002	usrt	1/22/2016 6:06:53 PM	DANIEL
1/14/2015	1/14/2015	1/14/2015	SPK-000001	Fiscal 2015	Sales order	000002	USPI	5637182092			Current	IV-8002	uspi	1/15/2015 1:02:46 PM	SUSAN
10/30/2016				Fiscal 2016		000002	GBSI	5637182092			Current	IV-8002	gsbi	1/18/2017 6:57:59 PM	JUNE
10/1/2016	10/1/2016	10/1/2016	2000001	Fiscal 2016	Purchase order	000002	FRRT	5637182092			Current	IV-8002	frst	2/7/2017 10:22:02 PM	Emmah
1/16/2016				Fiscal 2016		000002	FRSI	5637206852			Current	IV-8002	frsi	2/8/2017 10:21:27 PM	JUNE
1/15/2016	1/15/2016			Fiscal 2016	Project	000002	USPI	5637145350			Current	IV-8002	uspi	1/12/2016 6:41:42 PM	JUNE
1/15/2015	1/15/2015	1/15/2015	SPK-000002	Fiscal 2015	Sales order	000003	USMF	5637145350			Current	IV-8002	usmf	1/15/2015 5:25:43 PM	SIMON
1/15/2015	1/15/2015	1/15/2015	SPK-000002	Fiscal 2015	Sales order	000003	DEMIF	5637145350			Current	IV-8002	demif	1/16/2015 7:16:35 PM	SUSAN
1/21/2015	1/21/2015	1/21/2015	1000003	Fiscal 2015	Purchase order	000003	USRT	5637145350			Current	IV-8002	usrt	1/22/2016 6:07:26 PM	DANIEL

Figura 4.1: Tabella General Journal Entry

Allocation	AssetLeasePo...	AssetLea...	GeneralJournalEntry	IsCorrection	IsCredit	LedgerAccount	LedgerDimension	MainAccount	OriginalAc...	Payme...	PostingType	Projct SA	Quantity	ReasonRef	Reporting...	Text	Transactio
43.91	0	Lease asset	68719550946			201175--	201175--	201175	0	0	Payroll		0.00	0		-43.91	-43.91
14.42	0	Lease asset	68719551054			201175--	201175--	201175	0	0	Payroll		0.00	0		-14.42	-14.42
43.91	0	Lease asset	68719551191			201175--	201175--	201175	0	0	Payroll		0.00	0		-43.91	-43.91
14.42	0	Lease asset	68719551295			201175--	201175--	201175	0	0	Payroll		0.00	0		-14.42	-14.42
43.91	0	Lease asset	68719551427			201175--	201175--	201175	0	0	Payroll		0.00	0		-43.91	-43.91
1429.28	0	Lease asset	68719551602			201175--	201175--	201175	0	0	Payroll		0.00	0		-1429.28	-1429.28
73.42	0	Lease asset	68719551340			201175--	201175--	201175	0	0	Payroll		0.00	0		-73.42	-73.42
25,108.11	0	Lease asset	68719551342			201175--	201175--	201175	0	0	Payroll		0.00	0		-25,108.11	-25,108.11
122.54	0	Lease asset	68719551412			201180--	201180--	201180	0	0	Payroll		0.00	0		-122.54	-122.54
157.32	0	Lease asset	68719551413			201180--	201180--	201180	0	0	Payroll		0.00	0		-157.32	-157.32
112.07	0	Lease asset	68719551415			201180--	201180--	201180	0	0	Payroll		0.00	0		-112.07	-112.07
139.96	0	Lease asset	68719551531			201180--	201180--	201180	0	0	Payroll		0.00	0		-139.96	-139.96
122.53	0	Lease asset	68719551712			201180--	201180--	201180	0	0	Payroll		0.00	0		-122.53	-122.53
157.31	0	Lease asset	68719551713			201180--	201180--	201180	0	0	Payroll		0.00	0		-157.31	-157.31
112.08	0	Lease asset	68719551715			201180--	201180--	201180	0	0	Payroll		0.00	0		-112.08	-112.08
139.94	0	Lease asset	68719551547			201180--	201180--	201180	0	0	Payroll		0.00	0		-139.94	-139.94
122.54	0	Lease asset	68719551584			201180--	201180--	201180	0	0	Payroll		0.00	0		-122.54	-122.54
157.31	0	Lease asset	68719551585			201180--	201180--	201180	0	0	Payroll		0.00	0		-157.31	-157.31
112.07	0	Lease asset	68719551587			201180--	201180--	201180	0	0	Payroll		0.00	0		-112.07	-112.07

Figura 4.2: Tabella General Journal Account Entry

Il contenuto delle due tabelle è stato importato in un database locale creato e gestito tramite Microsoft SQL Server Management Studio. In aggiunta alle tabelle citate, sono stati inclusi anche ulteriori dati necessari per classificare ogni transazione in una specifica attività di chiusura contabile.

Quindi, è stato innanzitutto identificato un elenco delle principali attività di chiusura contabile mensile che un'azienda deve svolgere. Di seguito vengono elencate tali attività (tra parentesi viene riportato un identificativo univoco):

- **Registrare fatture fornitore (ACT01):** coinvolge la registrazione delle fatture dei fornitori nel sistema per mantenere una traccia accurata delle passività aziendali e garantire una corretta registrazione contabile.
- **Registrare e accoppiare tutti i pagamenti fornitori (ACT03):** coinvolge la registrazione e la finalizzazione di tutte le transazioni finanziarie. Durante questa fase di chiusura, vengono contabilizzati e registrati tutti i pagamenti effettuati dai fornitori, garantendo che le transazioni siano adeguatamente documentate nei registri finanziari. Questo processo include la verifica e l'adempimento di tutti gli obblighi di pagamento, l'aggiornamento dei saldi e la registrazione accurata delle transazioni nel sistema contabile.
- **Eseguire la rivalutazione monetaria (ACT04):** comporta l'esecuzione di una rivalutazione valutaria per adeguare i valori contabili alle variazioni dei tassi di cambio. Durante questa operazione, vengono riesaminati e aggiornati i valori delle transazioni in valute estere per riflettere le fluttuazioni del mercato valutario. Questo processo è cruciale per mantenere l'accuratezza e la coerenza dei dati contabili, assicurando che gli asset e gli obblighi siano valutati in modo corretto sulla base degli attuali tassi di cambio.
- **Registrare fatture di vendita (ACT07):** la registrazione delle fatture di vendita nel sistema contabile aziendale è un processo fondamentale per documentare le transazioni commerciali con i clienti. Le fatture di vendita sono essenziali per garantire una registrazione accurata delle entrate aziendali e per riflettere correttamente l'attività economica dell'azienda nei dati contabili.
- **Registrare e accoppiare tutti i pagamenti clienti (ACT09):** questa attività coinvolge la registrazione e la regolazione di tutti i pagamenti ricevuti dai clienti. Dopo aver ricevuto i fondi dai clienti, vengono creati i documenti contabili necessari, come registrazioni di giornale o altri documenti finanziari. Successivamente, si verifica e si concilia l'accuratezza delle informazioni finanziarie per garantire che i dati contabili riflettano correttamente l'entrata di denaro. Il processo include anche la gestione di eventuali differenze o discrepanze tra i dati contabili e i registri finanziari effettivi, assicurando un saldo corretto e aggiornato dei conti.

- **Eseguire la riconciliazione bancaria (ACT13):** viene eseguita la riconciliazione bancaria per garantire l'allineamento tra i dati contabili aziendali e i registri bancari. Questo processo coinvolge la comparazione accurata delle transazioni bancarie, come addebiti e accredito, con le registrazioni contabili dell'azienda. Vengono identificate e risolte eventuali discrepanze o differenze tra i saldi bancari e quelli contabili.
- **Registrare acquisizioni e correzioni di cespiti (ACT15):** questa attività implica la registrazione di tutte le acquisizioni o eventuali aggiustamenti provenienti da varie fonti come inventario, progetti, ecc., qualora siano necessari. Durante questa fase, vengono creati e contabilizzati i documenti appropriati per riflettere le modifiche nei dati patrimoniali dell'azienda. Ciò potrebbe includere l'aggiornamento di valori dell'inventario, la registrazione di costi aggiuntivi o di correzioni contabili necessarie. L'obiettivo è garantire che le informazioni contabili riflettano con precisione le transazioni legate alle acquisizioni o agli aggiustamenti provenienti da diverse aree aziendali.
- **Pubblicare tutte le vendite/scarti di beni (ACT16):** coinvolge la registrazione di tutte le vendite o delle operazioni di rottamazione di attività aziendali. Durante questo processo, vengono creati e contabilizzati i documenti appropriati per riflettere le transazioni relative alla cessione o alla rottamazione degli asset. Ciò include la registrazione di entrate derivanti dalle vendite di asset o la registrazione di eventuali perdite o guadagni derivanti dalla rottamazione.
- **Eseguire l'ammortamento dei cespiti del mese (ACT17):** viene eseguita la procedura di ammortamento per il mese corrente. Durante questo processo, il sistema contabile calcola e registra la quota di deprezzamento relativa agli asset aziendali soggetti a questo ammortamento periodico. Ciò può coinvolgere l'applicazione di metodi di ammortamento come il metodo dritto o il metodo delle quote crescenti, in base alle normative contabili e alle politiche aziendali. Garantisce che la contabilità aziendale rifletta in modo preciso il consumo di valore degli asset nel corso del tempo, mantenendo la conformità normativa e offrendo una visione accurata della situazione finanziaria dell'azienda.
- **Dichiarazione delle tasse (ACT29):** implica la dichiarazione delle imposte, un processo fondamentale per assicurarsi che l'azienda sia conforme alle normative fiscali e paghi le tasse dovute in modo accurato e tempestivo. Questa attività è essenziale per mantenere la trasparenza finanziaria, rispettare gli obblighi fiscali e mitigare il rischio di sanzioni fiscali.
- **Registrare le scritture di assestamento (ACT32):** coinvolge la registrazione di tutti gli aggiustamenti contabili necessari nel libro mastro dell'azienda. Gli aggiustamenti contabili sono modifiche apportate ai dati finanziari per correggere

errori, riflettere cambiamenti nelle condizioni finanziarie o adattarsi a nuove informazioni, è fondamentale per garantire l'integrità e l'accuratezza dei dati finanziari aziendali, permettendo di riflettere correttamente le transazioni e le circostanze che possono emergere nel corso dell'attività aziendale.

- **Registrare le allocazioni (ACT34):** consiste nella registrazione di tutte le allocazioni contabili necessarie nel sistema aziendale. Le allocazioni contabili vengono effettuate per distribuire i costi o i ricavi tra diverse entità o divisioni aziendali, al fine di assegnare correttamente le risorse finanziarie, è essenziale per garantire una distribuzione equa dei costi e dei ricavi tra le varie parti dell'azienda.
- **Registrare tutti i ratei e risconti (ACT35):** coinvolge la registrazione di tutti gli accantonamenti (accruals) contabili necessari. Gli accantonamenti rappresentano stime di costi o ricavi futuri non ancora registrati nel sistema contabile. Sono fondamentali per anticipare correttamente i costi o i ricavi futuri.
- **Eseguire la chiusura dell'anno fiscale (ACT44):** la chiusura dell'anno fiscale è un processo cruciale che comprende una serie di passaggi essenziali per consolidare i dati finanziari dell'azienda e prepararla per il nuovo anno fiscale. È di vitale importanza per garantire la precisione delle informazioni finanziarie aziendali, rispettare le normative contabili e fornire una base solida per la gestione finanziaria nel futuro.

Successivamente, è stata associata ciascuna transazione a una delle attività sopra elencate, basandosi principalmente su due campi presenti nelle tabelle menzionate in precedenza:

- **Posting Type:** indica il tipo di registrazione relativo al record considerato. Ad esempio, può indicare se il record corrisponde a una fattura di vendita, a un acquisto, a una transazione di payroll, a un registro di beni aziendali (cespiti), e così via. Il suddetto campo è presente all'interno della tabella GeneralJournalAccountEntry.
- **Journal Category:** indica la categoria o il tipo di libro giornale in cui è stata registrata la transazione. Ad esempio, questo campo potrebbe indicare se la transazione è relativa alle vendite, agli acquisti, alle correzioni o ad altre attività contabili specifiche. Il suddetto campo è presente all'interno della tabella GeneralJournalEntry.

Per completare il processo di mappatura delle attività di chiusura contabile, sono state identificate tutte le possibili combinazioni, esattamente 134, di Posting Type e Journal Category. Per ciascuna di queste combinazioni, è stata associata una delle attività precedentemente elencate. Il dataset finale è rappresentato da un file .csv generato attraverso una query SQL. Esso è composto da un totale di 8.327.369 transazioni, che coprono il periodo dal 01-01-2020 al 30-04-2023. Per ogni riga del dataset sono presenti 24 colonne, di seguito sono elencate le colonne di maggiore rilevanza:

- **RecId**: identificatore univoco di ogni transazione.
- **CreatedDateTime**: giorno, mese, anno e ora di creazione di una transazione.
- **SubledgerVoucherDataAreaId**: identificatore della sede di una filiale (Italia, Inghilterra, Francia, Polonia e Romania).
- **JournalCategory**: categoria o il tipo di libro giornale in cui è stata registrata la transazione.
- **PostingTypeLabel**: tipo di registrazione relativo al record considerato.
- **PostingTypeValue**: identificatore univoco del Posting Type.
- **CreatedBy**: nome dell'utente che ha creato la transazione.
- **ClosingActivityDescription**: descrizione dell'attività di chiusura contabile relativa alla transazione.
- **ClosingActivityId**: identificatore univoco per ogni attività di chiusura contabile.

## 4.2 Preprocessing ed Exploratory Data Analysis

Questa sezione si focalizza sull'analisi dettagliata del dataset utilizzato e descrive il processo per ottenere la serie temporale che sarà oggetto di previsione. Per ottenere una visione più dettagliata della distribuzione delle transazioni tra le diverse sedi aziendali, ho generato un istogramma che visualizza il numero di transazioni registrate per ciascuna company. Questo grafico permette di valutare in modo chiaro e diretto la quantità di attività svolta da ogni filiale, fornendo informazioni utili per analizzare eventuali disparità o tendenze rilevanti nell'ambito delle transazioni aziendali.

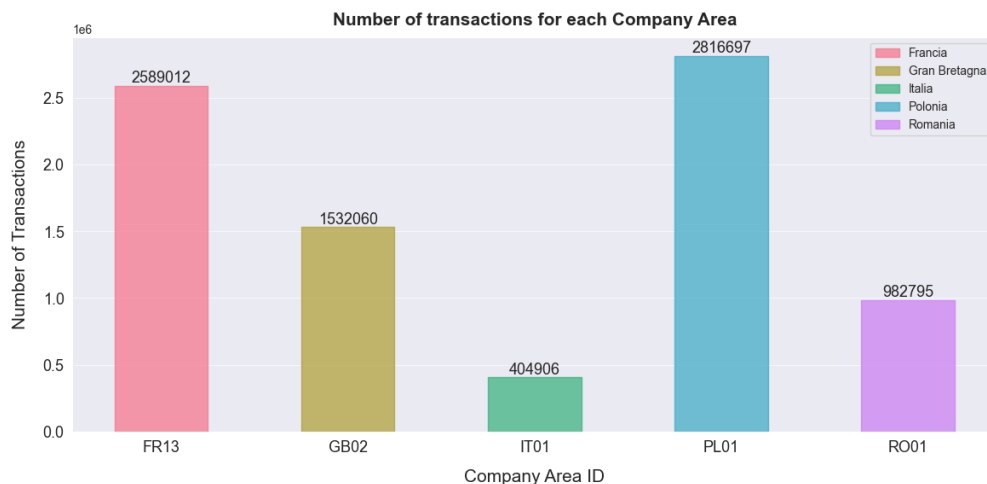


Figura 4.3: Istogramma relativo al numero di transazioni per ogni filiale da gennaio 2020 ad aprile 2023

La figura 4.3 ha rivelato interessanti pattern riguardo al volume di transazioni registrate dalle diverse sedi aziendali. In particolare, è emerso che le filiali situate in Francia e in Polonia presentano un volume di transazioni molto simile, indicando una consistente attività contabile in entrambi i paesi. Al contrario, le sedi in Gran Bretagna, Romania e soprattutto in Italia hanno mostrato un volume di transazioni nettamente inferiore. Questo fenomeno potrebbe essere attribuito a una serie di fattori, tra cui differenze nelle dimensioni e nella complessità delle operazioni aziendali, nelle pratiche contabili locali e nelle politiche di gestione finanziaria. Ad esempio, la Francia e la Polonia potrebbero avere procedure contabili più complesse o una maggiore attività commerciale che richiede un maggiore numero di transazioni per registrare accuratamente tutte le attività finanziarie. Al contrario, l'Italia potrebbe avere una struttura aziendale più snella o adottare pratiche contabili semplificate che richiedono un minor numero di transazioni per chiudere il bilancio mensile.

Un altro aspetto rilevante da considerare riguarda le transazioni generate tramite processi batch, ossia operazioni automatizzate per la registrazione di transazioni ripetitive che devono essere registrate regolarmente. Questi processi sono fondamentali per garantire l'efficienza operativa e la coerenza dei dati contabili, soprattutto per attività ricorrenti come addebiti mensili, pagamenti periodici e altre transazioni standardizzate. Attraverso il campo *CreatedBy*, ho identificato tutte le transazioni generate dall'utente "Admin", presumibilmente associate ai processi automatizzati appena citati.

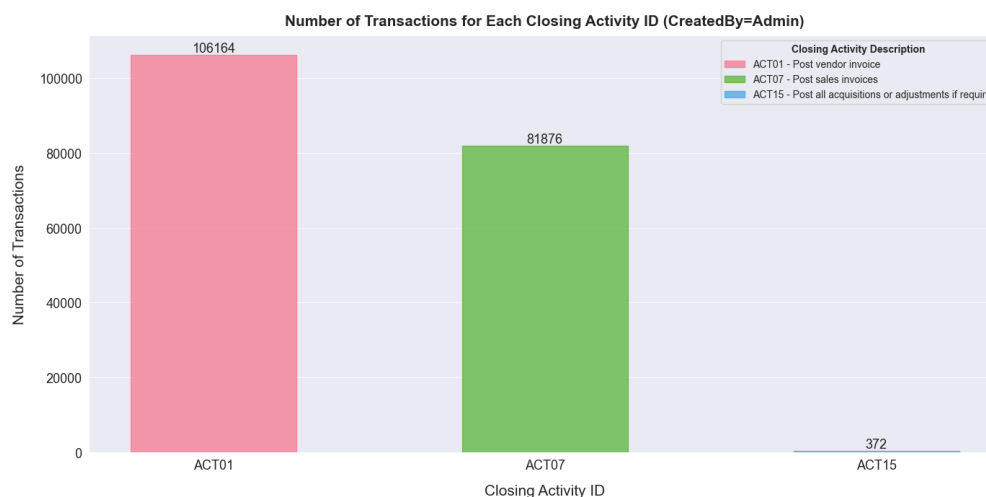


Figura 4.4: Istogramma relativo al numero di transazioni create dall'utente Admin

Come evidenziato nella Figura 4.4, si osserva che le principali attività influenzate da questi processi automatizzati sono quelle relative alla registrazione delle fatture di vendita e dei fornitori. Il numero totale di transazioni associate a tali attività ammonta a 188.412 su un totale di 8.327.369, rappresentando circa il 2% del dataset. Pertanto, si è deciso di mantenere questi record, poiché non hanno un impatto significativo sul complessivo volume di transazioni dato che richiedono comunque un impegno lavorativo, seppur

minore rispetto ad altre transazioni.

Infine, è stato utile visualizzare un altro istogramma che rappresenta il numero di transazioni per ogni singola attività di chiusura contabile. Questo grafico fornisce un'ulteriore visione d'insieme sulle attività svolte e sulla loro distribuzione nel dataset. Attraverso questa rappresentazione, è possibile individuare eventuali attività che presentano un volume di transazioni significativamente superiore o inferiore rispetto alle altre.



Figura 4.5: Istogramma relativo al numero di transazioni per ogni singola attività di chiusura contabile

La figura 4.5 mette in luce la diversa rilevanza delle varie attività nel contesto della chiusura contabile aziendale. Emergono chiaramente attività come la registrazione di scritture di assestamento, che presentano un impatto significativo nel numero totale di transazioni. Al contrario, altre attività, come la vendita dei beni aziendali, la dichiarazione delle tasse o la registrazione dei ratei e risconti, mostrano un impatto molto più esiguo sul volume complessivo delle transazioni.

In considerazione di ciò, si è presa la decisione di eliminare tutti i record associati alle attività identificate con i valori di ACT16, ACT29 e ACT35 nel campo *ClosingActivity-Id*. Questa scelta è stata motivata dal fatto che tali attività presentano un contributo trascurabile alla chiusura contabile e possono essere considerate marginali ai fini del processo di previsione delle serie temporali.

Per quanto riguarda la fase di preprocessing dei dati, è stata eseguita un'operazione di aggregazione delle transazioni al fine di ottenere una rappresentazione giornaliera del numero di transazioni per ciascuna filiale. Questo processo è stato fondamentale per trasformare i dati originali, che comprendevano transazioni individuali, in serie temporali aggregate che potessero essere utilizzate per l'analisi e la previsione.

L'obiettivo principale di questa trasformazione è stato quello di creare un dataset adatto per lo sviluppo di modelli di previsione temporale, consentendo di identificare i pattern e le tendenze nei volumi delle transazioni nel tempo. In questo modo, è possibile ottenere



una visione più chiara e informativa sulla distribuzione delle attività di chiusura contabile tra le diverse company e sui loro trend temporali.

Inizialmente, l'approccio adottato prevedeva la raccolta delle transazioni su base giornaliera. Tuttavia, questa scelta è stata rivista a seguito dell'identificazione di alcune criticità nell'aggregazione giornaliera dei dati. Infatti, la raccolta giornaliera delle transazioni ha portato alla creazione di serie temporali caratterizzate da numerosi valori nulli e picchi anomali che si discostavano significativamente dalla media del numero di transazioni. Per superare queste difficoltà e ottenere una rappresentazione più stabile e significativa dei dati, si è optato per la raccolta settimanale delle transazioni, utilizzando il lunedì come giorno di riferimento per definire ciascuna settimana. Questo approccio ha permesso di ridurre il rumore nei dati e di ottenere serie temporali più coerenti e informative, in grado di evidenziare in modo più accurato le tendenze e i pattern nel numero di transazioni nel tempo.

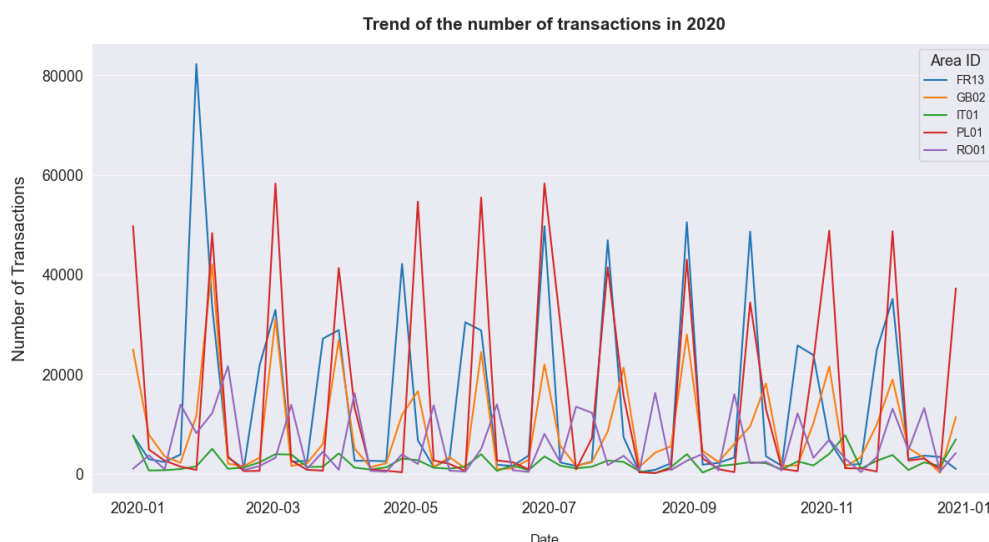


Figura 4.6: Trend del numero di transazioni nel corso del 2020

Nonostante l'aggregazione settimanale dei dati, il grafico mostra un andamento non uniforme nel numero di transazioni per ogni filiale nel corso del 2020, come evidenziato nella figura 4.6. Si notano picchi che si verificano principalmente alla fine di ogni mese, coincidenti con i giorni di chiusura contabile mensile, i quali risultano essere momenti di maggiore attività. È interessante osservare che tali picchi si presentano in settimane diverse per ogni filiale, suggerendo eventuali ritardi nella chiusura mensile o differenze nelle politiche aziendali tra le diverse sedi.

Inoltre, se esaminiamo il trend del numero di transazioni settimanali nel corso del 2020 per la filiale italiana, emerge un quadro caratterizzato dall'assenza di un pattern definito e regolare. La figura 4.7 illustra chiaramente questa situazione, evidenziando una variabilità significativa nel numero di transazioni registrate settimanalmente. Tale mancanza di un pattern ben definito suggerisce la presenza di fluttuazioni nell'attività aziendale,

potenzialmente influenzate da una serie di fattori esterni o interni all'azienda.

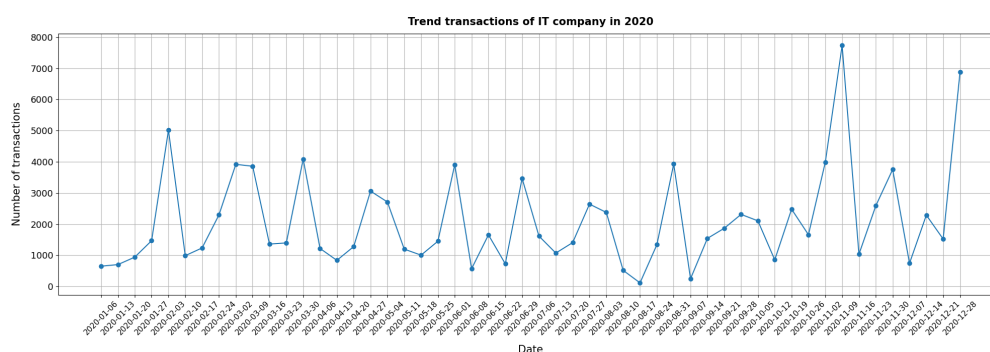


Figura 4.7: Trend del numero di transazioni nel corso del 2020 per la filiale Italia

Per supportare ulteriormente le affermazioni precedenti, è stata eseguita un'analisi di decomposizione additiva della serie temporale relativa alla filiale italiana, presa come esempio. Questa decomposizione consente di identificare i componenti principali della serie, come il trend, la stagionalità e i residui, offrendo un'ulteriore visione della variabilità dei dati nel tempo. Il risultati vengono mostrati nella figura 4.8.

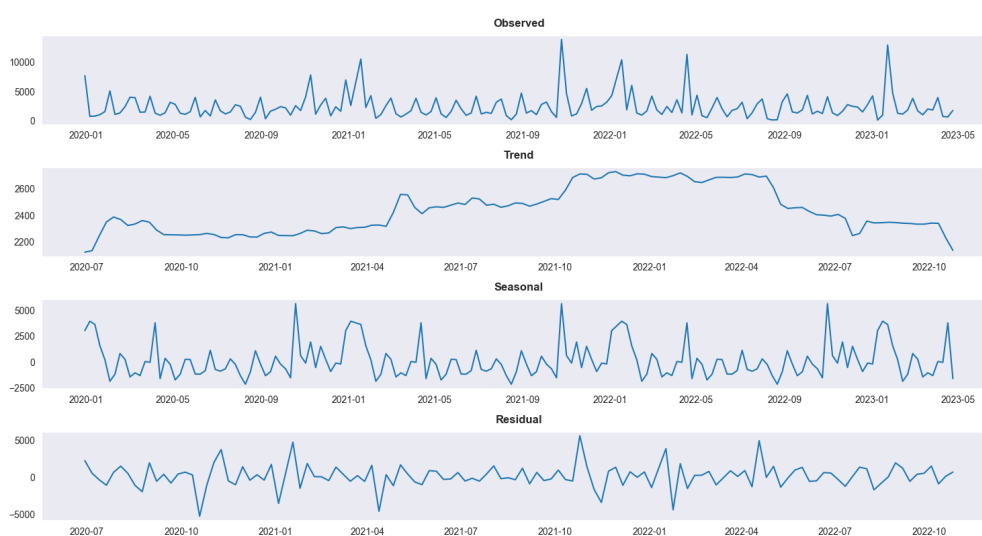


Figura 4.8: Decomposizione additiva della serie temporale

Per quanto riguarda la componente relativa al trend, si osserva un andamento sostanzialmente simile per la maggior parte della serie temporale, con un'eccezione evidente nel periodo compreso tra ottobre 2021 e maggio 2022, durante il quale si registra un significativo innalzamento del numero di transazioni. Questo fenomeno potrebbe essere indicativo di eventi specifici o cambiamenti nell'attività aziendale che hanno influenzato il volume delle transazioni in quel periodo. Mentre per la componente stagionale, non emerge un andamento regolare nel corso del tempo, confermando le informazioni precedenti riguardanti la mancanza di un pattern ben definito. Infine, l'analisi dei residui

evidenzia la presenza di un rumore significativo nella serie temporale, il che potrebbe influenzare le prestazioni di un modello statistico da adottare per le previsioni.

Un ulteriore aspetto da considerare, come già menzionato nei capitoli precedenti, riguarda la stazionarietà di una serie temporale. Questa caratteristica è fondamentale per garantire previsioni accurate, poiché una serie stazionaria mantiene costanti proprietà statistiche come media e varianza nel tempo. Pertanto, è stato eseguito il test statistico Augmented Dickey-Fuller (ADF) per verificare la stazionarietà delle serie temporali relative a ciascuna filiale.

Dopo aver condotto il test su ciascuna serie temporale, si valuta il valore del p-value. Se questo valore risulta inferiore o uguale a 0.05, si rigetta l'ipotesi nulla e si accetta l'ipotesi alternativa, il che indica la stazionarietà della serie temporale.

Company	Statistica ADF	p-value	Stazionarietà
Italia	-13.345	5.801e-25	SI
Francia	-4.057	0.0011	SI
Gran Bretagna	-3.1798	0.0211	SI
Romania	-4.0352	0.0012	SI
Polonia	-11.0622	4.777e-20	SI

Tabella 4.1: Risultati test di stazionarietà

La tabella 4.1 evidenzia che i valori del p-value sono tutti nettamente inferiori a 0.05, indicando una stazionarietà significativa per le serie temporali di tutte le filiali, ad eccezione della Gran Bretagna, dove il valore del p-value si avvicina a 0.05. Questo suggerisce che, mentre le serie temporali delle altre filiali mostrano una chiara stazionarietà, quella della Gran Bretagna potrebbe avere una stazionarietà meno evidente o potrebbe essere influenzata da fattori che richiedono ulteriori analisi.

Per completare il processo di preprocessing, sono state aggiunte alcune caratteristiche aggiuntive per arricchire ogni osservazione delle serie temporali. Attualmente, il dataset è composto da tre semplici campi, come mostrato in figura 4.9:

- **item\_id**: identificativo della sede della filiale.
- **timestamp**: data dell'osservazione.
- **target**: numero di transazioni per quell'osservazione.

	item_id	timestamp	target
0	FR13	2019-12-30	7663.0
1	FR13	2020-01-06	2917.0
2	FR13	2020-01-13	2341.0
3	FR13	2020-01-20	3853.0
4	FR13	2020-01-27	82251.0

Figura 4.9: Prime cinque righe del dataset contenente le serie temporali

Successivamente, per preservare le informazioni relative alle attività di chiusura contabile, sono state aggiunte le caratteristiche corrispondenti a ciascuna attività. A tal fine, è stata creata una colonna per ogni attività, contenente il numero di transazioni registrate in quella data specifica. È importante sottolineare che la somma dei valori di queste colonne aggiuntive corrisponde al numero della variabile target. Questo processo ha arricchito ulteriormente il dataset, consentendo al modello di catturare dettagli significativi riguardanti le attività di chiusura contabile.

Dopo aver aggiunto queste caratteristiche, è stata svolta un'analisi della matrice di correlazione, una pratica fondamentale nell'esplorazione dei dati. Questa analisi fornisce informazioni cruciali sulla relazione tra le diverse variabili presenti nel dataset. Quando si lavora con serie temporali e più feature correlate, come nel caso delle diverse attività di chiusura contabile, eseguire questa analisi diventa particolarmente importante. La visualizzazione della matrice di correlazione consente di identificare eventuali relazioni lineari tra le variabili, rivelando se e quanto due o più variabili si muovono insieme. Nel contesto delle attività di chiusura contabile, una correlazione positiva tra due attività potrebbe indicare che l'aumento o la diminuzione del numero di transazioni in una attività è associato a un comportamento simile nell'altra attività. Al contrario, una correlazione negativa suggerirebbe un movimento opposto tra le attività. La figura 4.10 mostra la matrice di correlazione ottenuta.

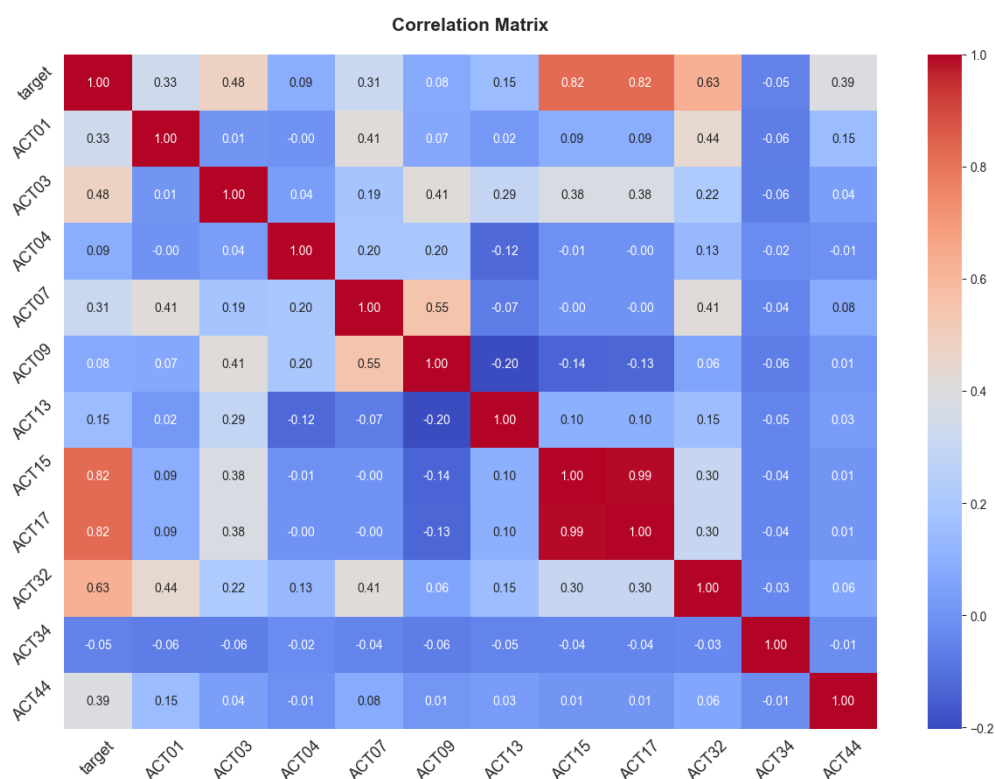


Figura 4.10: Matrice di correlazione

Il grafico evidenzia una forte correlazione tra le attività identificate rispettivamente dal

codice ACT15 e ACT17. Queste attività, che riguardano la registrazione e l'acquisizione di cespiti (ACT15) e l'ammortamento dei cespiti stessi (ACT17), mostrano un coefficiente di correlazione pari a 0.99. Anche solitamente si opta per scartare una delle due feature quando sono altamente correlate, poiché potrebbero non aggiungere informazioni significative alla variabile obiettivo delle previsioni, abbiamo deciso di mantenere entrambe le attività. Questa decisione è stata presa considerando che la dimensionalità delle feature non è elevata, quindi il rischio di sovraccaricare il modello con troppe informazioni è limitato.

Infine, considerando che il volume di transazioni di chiusura contabile varia notevolmente tra le diverse filiali (figura 4.3), si è scelto di arricchire ulteriormente la serie temporale aggiungendo una nuova colonna. Questa colonna rappresenta il logaritmo naturale dei valori della variabile *target*, ovvero il numero di transazioni. Aggiungere questa feature è una tecnica comune nelle previsioni di serie temporali, soprattutto quando si hanno valori che variano su un'ampia gamma o che sono distribuiti in modo non uniforme. L'applicazione del logaritmo ai valori delle feature può aiutare a stabilizzare la varianza dei dati e a ridurre la distorsione causata da valori estremi o anomalie. In particolare, l'uso del logaritmo può essere utile quando i dati hanno una distribuzione asimmetrica o quando sono fortemente influenzati da valori estremi. Utilizzando il logaritmo dei valori, viene "appiattita" la distribuzione dei dati, rendendoli più simmetrici e facili da trattare per molti algoritmi di forecasting. Questo può portare a modelli più stabili e generalizzabili. Di seguito vengono mostrate le prime cinque righe del dataset utilizzando per le prossime fasi di implementazione del modello di forecasting.

	item_id	timestamp	target	ACT01	ACT03	ACT04	ACT07	ACT09	ACT13	ACT15	ACT17	ACT32	ACT44	log_target
0	FR13	2019-12-30	7663.0	1267.0	299.0	0.0	2108.0	189.0	84.0	0.0	0.0	3716.0	0.0	8.944159
1	FR13	2020-01-06	2917.0	1052.0	490.0	2.0	103.0	255.0	191.0	3.0	0.0	821.0	0.0	7.978311
2	FR13	2020-01-13	2341.0	846.0	115.0	0.0	280.0	69.0	66.0	416.0	407.0	142.0	0.0	7.758333
3	FR13	2020-01-20	3853.0	1350.0	932.0	44.0	568.0	297.0	278.0	81.0	80.0	223.0	0.0	8.256607
4	FR13	2020-01-27	82251.0	1871.0	863.0	0.0	609.0	363.0	219.0	2.0	0.0	19393.0	58931.0	11.317531

Figura 4.11: Prime cinque righe del dataset finale contenente le serie temporali

### 4.3 Model selection

In questa sezione verranno esaminate e valutate diverse opzioni di algoritmi di previsione delle serie temporali per il problema in questione. Questa fase è cruciale poiché determinerà quale modello sarà utilizzato per effettuare le previsioni future. Saranno esplorate diverse metodologie per la selezione del modello ottimale, valutando le prestazioni attraverso criteri appropriati.

In primo luogo, è stata presa una decisione fondamentale, ovvero quella di stabilire l'orizzonte temporale da prevedere. Si è optato per un orizzonte di previsione di 12 settimane, corrispondenti a circa tre mesi. Tale scelta è stata motivata da diverse considerazioni.

Innanzitutto, un orizzonte temporale di tre mesi fornisce una prospettiva sufficientemente ampia per comprendere le tendenze a lungo termine e catturare eventuali pattern stagionali o ciclici nelle serie temporali. Inoltre, un periodo di tre mesi è comunemente utilizzato in contesti aziendali per pianificare attività e risorse a breve termine. Questo intervallo di tempo è considerato abbastanza flessibile da permettere una previsione accurata delle fluttuazioni nel numero di transazioni senza compromettere eccessivamente la precisione del modello. Una volta presa questa scelta, il dataset è stato suddiviso in due parti: il training set e il test set. Per suddividere il dataset in un training set e un test set, è stata adottata un approccio temporale. Prima di tutto, è stata identificata la data massima presente nel dataset, che rappresenta il punto temporale più recente disponibile. Utilizzando questa data massima, è stata calcolata la data di inizio del test set, retrocedendo di 12 settimane, in modo da garantire un periodo sufficiente di dati futuri per il test. Infine, il dataset è stato suddiviso in base alla data di inizio del test set: tutte le osservazioni fino a quella data sono state assegnate al training set, mentre le osservazioni successive hanno costituito il test set. Questo approccio temporale assicura che il modello venga addestrato su dati storici precedenti e testato su dati futuri, simulando così un ambiente realistico di previsione. Di seguito è riportato il codice Python utilizzato per eseguire la suddivisione del dataset in un training set e un test set.

```
prediction_length = 12 # Length of the prediction

# Convert the 'timestamp' column to datetime format if it's not already
# in that format
df['timestamp'] = pd.to_datetime(df['timestamp'])

# Find the maximum date in the dataset
max_date = df['timestamp'].max()

# Calculate the start date of the test set
test_start_date = max_date - pd.DateOffset(weeks=prediction_length+1)

# Split the dataset into training and test sets
train_data = df[df['timestamp'] <= test_start_date]
test_data = df[df['timestamp'] > test_start_date]
```

Listing 4.1: Suddivisione del dataset in training set e test set

Dopodiché, si è concentrata l'attenzione sulla selezione della metrica da minimizzare durante la fase di addestramento del modello. La scelta è ricaduta sul Mean Absolute Scaled Error (MASE) [18], la quale offre un'interpretazione intuitiva della precisione delle previsioni, poiché si basa sull'errore assoluto medio (MAE) e confronta le previsioni del modello con quelle di un modello di riferimento "naive". Un valore di MASE inferiore a 1 indica una maggiore precisione rispetto al modello naive, mentre un valore superiore a 1 suggerisce prestazioni inferiori.

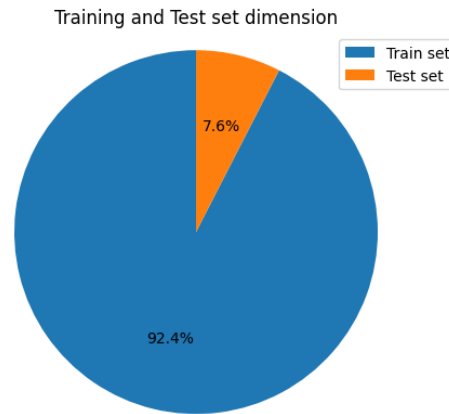


Figura 4.12: Porzioni del training set e del test set

Inoltre, il MASE (Mean Absolute Scaled Error) offre diversi vantaggi che lo rendono una scelta appropriata per valutare le prestazioni dei modelli di previsione:

- **Metrica normalizzata:** viene normalizzato l'errore per ciascuna serie temporale in base alla scala di quella serie temporale stessa. Questo significa che tiene conto delle diverse grandezze delle serie temporali, consentendo una valutazione più equa e comparabile delle prestazioni del modello su serie temporali con scale diverse.
- **Non sensibile alle serie costanti:** poiché il MASE è definito come il rapporto tra l'errore assoluto medio del modello in esame e l'errore medio assoluto di un modello di benchmark semplice, non è influenzato dalle serie temporali costanti che hanno un errore assoluto medio pari a zero.
- **Robusto agli outliers:** è meno sensibile agli outlier rispetto ad altre metriche, poiché si basa sull'errore assoluto medio anziché sull'errore quadratico medio (RMSE).

Per la selezione del modello più adatto, è stata utilizzata la libreria AutoGluon, che ha permesso di valutare diversi modelli disponibili. Prima di esaminarli, è importante comprendere come la libreria gestisce la fase di cross-validation [19]. Durante il processo di addestramento del modello, AutoGluon esegue automaticamente la cross-validation. In questa fase, il dataset di addestramento viene diviso in due parti: un insieme di addestramento e un insieme di validazione. Questo consente al sistema di valutare le prestazioni dei vari modelli sui dati di validazione. Per impostazione predefinita, il set di validazione consiste in una finestra temporale che include gli ultimi passaggi temporali di ciascuna serie nel dataset, in base alla scelta della lunghezza dell'orizzonte temporale da prevedere. Ad ogni modo, è possibile personalizzare questa configurazione specificando il numero desiderato di finestre di validazione. La finestra di validazione è stata impostata su 5 per consentire una valutazione più accurata delle prestazioni dei modelli durante la fase di addestramento. Questo significa che il set di dati di addestramento viene diviso in

5 parti, e in ciascuna iterazione, una di queste parti viene utilizzata come set di validazione mentre le restanti vengono utilizzate per addestrare il modello. Questo approccio permette di ottenere una valutazione robusta delle prestazioni del modello su dati non visti durante l'addestramento, contribuendo a ridurre il rischio di overfitting e garantendo una migliore generalizzazione del modello. Di seguito è mostrata un'immagine esplicativa della cross-validation.

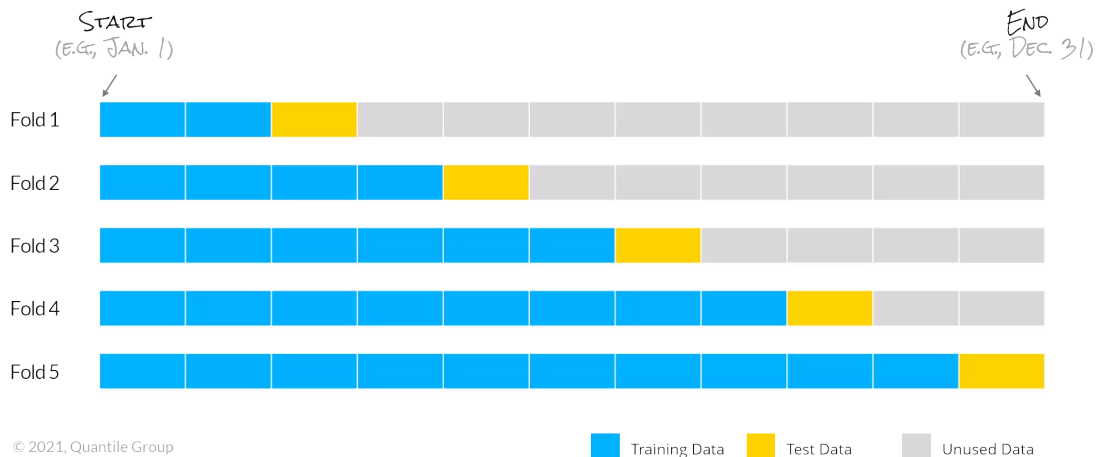


Figura 4.13: Cross-validation con numero di finestre di validazione pari a 5

All'interno della libreria AutoGluon sono disponibili diversi modelli, ognuno con caratteristiche e complessità diverse, che vanno dai modelli statistici ai modelli di deep learning. Viene fornita una breve descrizione per ogni modello valutato:

- **ETS:** il modello di smoothing esponenziale (ETS) [20] rappresenta un pilastro fondamentale nel campo della previsione temporale. Proposto per la prima volta alla fine degli anni '50 da Brown, Holt e Winters, l'ETS è stato ampiamente adottato grazie alla sua semplicità concettuale e alla sua efficacia nel generare previsioni accurate per una vasta gamma di serie temporali. L'approccio alla base dell'ETS si basa sull'idea di assegnare pesi esponenziali alle osservazioni passate, il che significa che le osservazioni più recenti hanno un peso maggiore rispetto a quelle più datate. Questo metodo consente al modello di reagire rapidamente ai cambiamenti nei dati e di adattarsi dinamicamente alle variazioni delle serie temporali nel tempo. Il modello ETS è composto da tre componenti principali: errore, tendenza e stagionalità. L'errore rappresenta la variabilità casuale nei dati che non segue alcun modello specifico, mentre la tendenza indica la direzione generale dei dati nel tempo, che può essere crescente, decrescente o costante. La stagionalità cattura i pattern periodici o ciclici nei dati che si ripetono a intervalli regolari, come le fluttuazioni stagionali nelle vendite al dettaglio o nelle previsioni meteorologiche. A seconda della presenza o meno di queste componenti, i modelli ETS possono essere suddivisi in diverse categorie. Alcuni modelli includono tutte e tre le componenti



(errore, tendenza e stagionalità), mentre altri possono omettere la stagionalità o persino la tendenza, a seconda delle caratteristiche specifiche dei dati. La principale attrattiva del modello ETS risiede nella sua capacità di generare previsioni affidabili in modo rapido e per una vasta gamma di serie temporali.

- **ARIMA:** il modello ARIMA, acronimo di Autoregressive Integrated Moving Average [21] è un altro metodo ampiamente utilizzato per la previsione delle serie temporali. La sua popolarità deriva dalla sua flessibilità e capacità di gestire una vasta gamma di pattern nei dati temporali. Introdotto per la prima volta negli anni '70 da George E. P. Box e Gwilym M. Jenkins, ARIMA combina tre componenti principali: autoregressione (AR), integrazione (I) e media mobile (MA). La componente di autoregressione riflette l'effetto della storia delle osservazioni sulla previsione futura, dove l'andamento futuro dipende linearmente dagli andamenti passati. La componente di integrazione si riferisce alla differenziazione dei dati per renderli stazionari, ovvero con media e varianza costanti nel tempo. Infine, la componente media mobile riflette l'effetto di un errore residuo su un'osservazione, dove l'errore è un'osservazione inattesa e casuale che non può essere spiegata dalle osservazioni passate. Il modello ARIMA è in grado di catturare una vasta gamma di pattern nei dati, inclusi trend lineari o non lineari, stagionalità e ciclicità. Questo lo rende una scelta versatile per la previsione di serie temporali in molte applicazioni diverse, dalle vendite al dettaglio alla finanza, alla meteorologia e oltre.
- **NPTS:** il modello NPTS [22], acronimo di "non-parametric baseline models for time series forecasting", è caratterizzato da un approccio non parametrico alla previsione delle serie temporali. A differenza dei modelli classici, che assumono una forma parametrica per la distribuzione predittiva, il modello NPTS genera previsioni campionando dalla distribuzione empirica secondo una strategia regolabile. Questo approccio consente al modello di produrre previsioni sempre ragionevoli, evitando così i problemi di stabilità numerica che possono verificarsi con alcuni modelli classici in presenza di distribuzioni di dati particolarmente complesse. Inoltre, il modello NPTS include una versione globale che sfrutta le informazioni provenienti da più serie temporali correlate per apprendere automaticamente la strategia di campionamento. Gli esperimenti empirici condotti dimostrano che il modello NPTS offre prestazioni ragionevoli e consistenti su una varietà di dataset, confermandone così l'efficacia come solida base di riferimento per la previsione delle serie temporali.
- **DeepAR:** progettato per la previsione probabilistica delle serie temporali, si basa sull'addestramento di un modello di rete neurale ricorrente autoregressivo su un ampio numero di serie temporali correlate [23]. L'obiettivo è stimare la distribuzione di probabilità futura di una serie temporale date le sue osservazioni passate.

Questo tipo di previsione probabilistica è essenziale in molte applicazioni commerciali, come nel settore della vendita al dettaglio, dove le previsioni di domanda probabilistiche sono fondamentali per ottimizzare la gestione delle scorte. La metodologia del modello si basa sull'applicazione di tecniche di deep learning alla previsione, con l'obiettivo di superare le sfide incontrate dai metodi classici ampiamente utilizzati. Il modello DeepAR è in grado di produrre previsioni più accurate rispetto ad altri metodi all'avanguardia, pur richiedendo un minimo sforzo manuale. Questo è dimostrato attraverso valutazioni empiriche su diversi dataset reali di previsione, dove DeepAR ha dimostrato di ottenere risultati superiori rispetto ad altri approcci.

- **PatchTST**: è conosciuto come Patch Time Series Transformer [24], rappresentando un'evoluzione significativa nel campo della previsione multivariata delle serie temporali e dell'apprendimento di rappresentazioni auto-supervisionate. La sua progettazione si basa su due pilastri fondamentali che ne caratterizzano l'efficienza e l'efficacia. In primo luogo, la suddivisione delle serie temporali in patch a livello di sotto serie rappresenta un'innovazione importante. Questo approccio consente di mantenere informazioni semantiche locali nell'embedding, migliorando così la capacità del modello di catturare dettagli rilevanti nelle serie temporali. Inoltre, la segmentazione delle serie in patch consente una riduzione significativa del consumo di memoria e di calcolo durante l'uso delle mappe di attenzione del Transformer. Questo vantaggio è particolarmente rilevante in applicazioni che richiedono la gestione di grandi volumi di dati temporali. Un'altra caratteristica chiave del modello è l'indipendenza dei canali. Ogni canale rappresenta una singola serie temporale univariata, ma condivide lo stesso embedding e i pesi del Transformer con le altre serie. Questo approccio permette al modello di apprendere rappresentazioni condivise tra serie diverse, migliorando la generalizzazione e consentendo al modello di estrarre pattern comuni da insiemi di dati eterogenei. L'architettura PatchTST si distingue per la sua capacità di migliorare significativamente l'accuratezza delle previsioni a lungo termine rispetto ai modelli Transformer tradizionali. Questo rende il modello particolarmente adatto a scenari in cui la previsione a lungo termine è cruciale, come ad esempio nella previsione delle vendite al dettaglio o nella previsione del traffico.
- **TemporalFusionTransformer**: per una descrizione dettagliata del seguente modello, si rimanda alla sezione 2.4 dedicata.
- **WeightedEnsemble**: nella libreria di AutoGluon [12], dopo che il sistema ha addestrato sequenzialmente i singoli modelli, viene utilizzato un algoritmo di selezione, detto forward selection, per combinare i risultati dei modelli individuali. Questo algoritmo seleziona iterativamente i modelli che contribuiscono maggiormente alla performance complessiva e costruisce un insieme pesato dei loro output. L'output

finale dell'ensemble è una combinazione convessa delle previsioni dei singoli modelli, dove i pesi dei modelli sono vincolati a essere non negativi e ad avere una somma uguale ad 1.

$$\hat{\mathbf{y}}_{i,T+1:T+H}^{\text{ensemble}} = \sum_{m=1}^M w_m \cdot \hat{\mathbf{y}}_{i,T+1:T+H}^{(m)} \quad \text{subject to } w_m \geq 0, \sum_{m=1}^M w_m = 1,$$

dove il secondo termine del prodotto all'interno sommatoria rappresenta le previsioni puntuali o i quantili generate da ciascuno degli  $M$  modelli addestrati. Questo approccio consente di ottenere una combinazione ottimale delle previsioni dei singoli modelli, che può essere utilizzata per generare previsioni puntuali o quantili.

#### 4.3.1 Risultati model selection

Tutti i modelli citati nel paragrafo precedente sono stati testati sullo stesso sottoinsieme di dati di training e tutti i risultati sono stati cross validati per evitare overfitting. Nella tabella 4.2 sono riportati i risultati relativi alla selezione del modello, insieme al tempo di addestramento espresso in secondi per ciascun modello.

Modello	MASE	Durata fitting del modello (s)
ETS	1,040	72
ARIMA	1,239	47
NPTS	0,861	0,2
DeepAR	0,996	224
PatchTST	0,741	136
TemporalFusionTransformer	<b>0,536</b>	660
WeightedEnsemble	0,536	7

Tabella 4.2: Risultati model selection

Emergono due modelli con MASE inferiore rispetto agli altri: il Temporal Fusion Transformer e il Weighted Ensemble, entrambi con lo stesso punteggio. Questo risultato suggerisce che il Weighted Ensemble abbia assegnato tutti i pesi al modello Temporal Fusion Transformer, come confermato dall'analisi dei log durante il fitting di quest'ultimo. È evidente che la combinazione con gli altri modelli non ha portato a miglioramenti della funzione di loss (MASE), dunque il Weighted Ensemble ha privilegiato l'uso del Temporal Fusion Transformer per ottenere le previsioni più accurate.

Dunque, alla luce dei risultati ottenuti, si è optato per l'utilizzo del modello Temporal Fusion Transformer per effettuare previsioni sulle serie temporali relative al quantitativo di lavoro necessario per la chiusura del bilancio mensile. Tuttavia, poiché durante la fase di selezione del modello sono stati utilizzati gli iperparametri predefiniti forniti

dalla libreria AutoGluon, nella prossima sezione verrà eseguita un'ottimizzazione degli iperparametri.

## 4.4 Ottimizzazione degli iperparametri

L'ottimizzazione degli iperparametri è il processo di selezione dei valori ottimali per gli iperparametri di un modello di machine learning al fine di massimizzare le prestazioni del modello su un insieme di dati di addestramento. Gli iperparametri sono parametri che non sono appresi direttamente dal modello durante il processo di addestramento, ma devono essere impostati esternamente prima dell'avvio del processo di addestramento.

Questo processo coinvolge solitamente l'esplorazione di un insieme di possibili combinazioni di valori per gli iperparametri al fine di trovare quella che massimizza una metrica di valutazione prestabilita. Questa esplorazione può essere eseguita in modo sistematico utilizzando tecniche come la ricerca esaustiva, la ricerca casuale, la ricerca bayesiana o l'ottimizzazione guidata da algoritmi di apprendimento.

L'obiettivo dell'ottimizzazione degli iperparametri è trovare una combinazione di valori per gli iperparametri che generalizzi bene sui dati non visti, migliorando così le prestazioni del modello e riducendo il rischio di overfitting o underfitting.

All'interno della libreria AutoGluon, sono disponibili diversi iperparametri configurabili per il modello Temporal Fusion Transformer. Questi iperparametri svolgono un ruolo cruciale nel plasmare il comportamento del modello e nella sua capacità di adattarsi ai dati di input. Viene fornito un elenco dei principali iperparametri che possono essere modificati e ottimizzati per migliorare le prestazioni del modello:

- **Context length:** definisce il numero di valori passati utilizzati per effettuare le previsioni future. In pratica, questo parametro determina quanti passi temporali precedenti il modello deve considerare quando genera una previsione per un dato istante nel tempo. Un valore più alto dell'iperparametro consente al modello di catturare meglio le tendenze a lungo termine e i pattern complessi presenti nei dati storici. Ciononostante, un valore troppo elevato potrebbe aumentare la complessità del modello e il tempo di calcolo necessario per l'addestramento e l'inferenza. D'altra parte, un valore più basso potrebbe essere utile quando si lavora con serie temporali ad alta frequenza o con cambiamenti repentini nei dati, consentendo al modello di reagire più rapidamente alle variazioni recenti (tipo int, valore di default =  $\max(64, 2 \cdot \text{prediction\_length})$ ).
- **Disable static features:** funzionalità che consente di gestire l'utilizzo delle caratteristiche statiche all'interno del modello di previsione. Quando impostato su *True*, questo parametro indica al modello di ignorare le caratteristiche statiche, anche se presenti nel dataset. Al contrario, quando impostato su *False*, il modello utilizzerà le caratteristiche statiche se presenti nel dataset. Le caratteristiche statiche sono quelle che non variano nel tempo per ciascuna serie temporale, come ad

esempio la categoria di un prodotto. Queste caratteristiche forniscono al modello informazioni aggiuntive che possono essere utili per migliorare la qualità delle previsioni (tipo booleano, valore di default = False).

- **Disable known covariates:** caratteristica che determina l'utilizzo delle covariate conosciute all'interno del modello di previsione. Quando impostato su True, questo parametro indica al modello di ignorare le covariate conosciute, anche se presenti nel dataset. Al contrario, quando impostato su False, il modello utilizzerà le covariate conosciute se presenti nel dataset. Le covariate conosciute sono variabili esterne che possono influenzare la serie temporale oggetto di previsione, ma che non variano nel tempo all'interno del dataset. Ad esempio, le covariate conosciute potrebbero comprendere informazioni demografiche, indicatori economici o segnalazioni dei giorni festivi (tipo booleano, valore di default = False).
- **Disable past covariates:** controlla l'utilizzo delle covariate passate nel modello. Le covariate passate si riferiscono a variabili esterne che sono state osservate in un periodo temporale precedente rispetto alla previsione attuale. Queste covariate possono essere informazioni storiche che influenzano il fenomeno che stiamo cercando di prevedere. Quando l'iperparametro è impostato su False, il modello utilizzerà queste covariate passate, se presenti nel dataset, per migliorare la precisione delle previsioni. Viceversa, se è impostato su True, il modello ignorerà completamente le covariate passate, indipendentemente dalla loro presenza nel dataset. Questo può essere utile quando si desidera escludere l'influenza delle covariate passate o quando si ritiene che non siano rilevanti per le previsioni future (tipo booleano, valore di default = False).
- **Hidden dim:** definisce la dimensione degli stati nascosti all'interno del modello, sia per la parte LSTM (Long Short Time Memory) che per quella transformer. Gli stati nascosti sono essenziali per il funzionamento del modello in quanto rappresentano la capacità di memorizzazione e apprendimento delle relazioni nei dati di input. Una dimensione maggiore degli stati nascosti consente al modello di catturare una maggiore complessità nei dati e di apprendere rappresentazioni più ricche delle relazioni temporali all'interno delle serie storiche. Un valore troppo elevato del parametro potrebbe portare a un'eccessiva complessità del modello, aumentando il rischio di overfitting e rendendo il processo di addestramento più lento e più computazionalmente costoso. Mentre una dimensione degli stati nascosti troppo bassa potrebbe limitare la capacità del modello di catturare informazioni importanti nei dati e di generare previsioni accurate. Pertanto, la scelta della dimensione degli stati nascosti dipende dalla complessità del problema di previsione e dalla disponibilità di risorse computazionali. È consigliabile eseguire esperimenti empirici con diverse dimensioni degli stati nascosti per determinare il valore ottimale che massimizza le prestazioni del modello (tipo int, valore di default = 32).

- **Variable dim:** definisce la dimensione degli embedding delle caratteristiche nel modello. Queste ultime sono rappresentazioni dense e a bassa dimensionalità delle variabili di input, che consentono al modello di apprendere rappresentazioni significative e compatte delle informazioni presenti nei dati. Aumentare la dimensione degli embedding delle caratteristiche permette al modello di acquisire una rappresentazione più ricca e dettagliata delle variabili di input, consentendo di catturare relazioni più complesse e sottili tra le caratteristiche. Allo stesso tempo, è importante bilanciare la dimensione del parametro in modo da evitare l'eccessivo aumento della complessità del modello e il rischio che il modello non generalizzi a sufficienza (tipo int, valore di default = 32).
- **Num heads:** specifica il numero di "heads" di attenzione nel decoder del modello. Nella struttura Transformer, l'auto-attention è una tecnica chiave utilizzata per catturare le relazioni a lungo raggio tra le parti della sequenza. Ogni head attention calcola un set di pesi che rappresenta l'importanza relativa delle diverse posizioni nella sequenza, consentendo al modello di concentrarsi su diverse parti durante l'encoding o decoding. La scelta del valore di questo parametro dipende dalla complessità del problema e dalle dimensioni del dataset. Solitamente, si preferisce un valore basso quando il dataset è di dimensioni ridotte, poiché riduce la complessità computazionale del modello. D'altra parte, un valore più alto può essere preferibile per dataset più complessi, in quanto consente al modello di catturare relazioni più intricate nella sequenza, migliorando la capacità di apprendimento (tipo int, valore di default = 4).
- **Dropout rate:** indica il tasso di dropout utilizzato per la regolarizzazione durante l'addestramento del modello. Il dropout è una tecnica di regolarizzazione comunemente utilizzata nei modelli neurali per prevenire l'overfitting. Durante l'addestramento, alcuni neuroni o connessioni vengono casualmente disattivati con una certa probabilità, definita proprio dal tasso di dropout. Se per esempio è impostato a 0.1, significa che il 10% dei neuroni verrà disattivato casualmente durante ciascun passaggio di addestramento. Questo parametro a migliorare la capacità di generalizzazione del modello, costringendo il modello a non fare affidamento eccessivo su singoli neuroni o connessioni specifiche. Di conseguenza, è importante bilanciare il valore del dropout rate in modo da evitare una regolarizzazione eccessiva, che potrebbe compromettere le prestazioni del modello, o una regolarizzazione insufficiente. Pertanto, il valore del dropout rate può essere regolato in base alle caratteristiche specifiche del dataset e alla complessità del problema (tipo float, valore di default = 0.1).
- **Max epochs:** specifica il numero massimo di epoche per le quali il modello verrà addestrato durante il processo di fitting. Un'epoca rappresenta un singolo passaggio attraverso l'intero dataset di addestramento durante il quale il modello

aggiorna i suoi pesi in base al calcolo dell'errore rispetto alle previsioni desiderate. Aumentare il numero di epoche può consentire al modello di apprendere ulteriori dettagli e pattern nei dati di addestramento, migliorando potenzialmente le prestazioni del modello. Un numero eccessivamente elevato di epoche potrebbe portare all'overfitting, dove il modello impara troppo bene i dettagli del dataset di addestramento e quindi perde la capacità di generalizzazione su dati non visti. Pertanto, è importante bilanciare il valore di `max_epochs` in modo da evitare l'overfitting e massimizzare le prestazioni del modello sui dati di test. Questo valore può essere regolato in base alla complessità del problema, alla dimensione del dataset e ad altri fattori specifici del contesto dell'applicazione (tipo `int`, valore di default = 100).

- **Batch size:** definisce la dimensione dei batch utilizzati durante il processo di addestramento del modello. Durante l'addestramento di una rete neurale, i dati vengono divisi in batch di dimensioni uguali e ciascuno viene utilizzato per calcolare l'errore e aggiornare i pesi del modello. L'utilizzo di batch consente di ridurre il tempo necessario per eseguire un singolo passaggio attraverso l'intero dataset di addestramento, poiché il modello viene aggiornato più frequentemente e i calcoli possono essere eseguiti in parallelo. Inoltre, l'uso di batch può contribuire alla stabilizzazione del processo di addestramento, poiché l'aggiornamento dei pesi viene eseguito su un insieme di dati rappresentativo anziché su singoli esempi. La scelta della dimensione del batch dipende da vari fattori, tra cui le risorse computazionali disponibili, la complessità del modello e la dimensione del dataset. Dimensioni del batch più grandi possono portare a una maggiore velocità di addestramento ma possono richiedere più memoria, mentre dimensioni del batch più piccole possono portare a una maggiore stabilità del modello ma richiedere più tempo per l'addestramento (tipo `int`, valore di default = 64).
- **Learning rate:** controlla la dimensione dei passi che il modello compie durante l'ottimizzazione dei pesi nella fase di addestramento. Un tasso di apprendimento più elevato consente al modello di convergere più rapidamente durante l'addestramento, ma può causare oscillazioni o saltare sopra minimi locali nella funzione di perdita. D'altra parte, un valore troppo basso può rallentare il processo di addestramento e far sì che il modello impieghi più tempo per convergere a una soluzione ottimale. La scelta della misura del parametro dipende dalla natura del problema, dalla complessità del modello e dalla quantità di dati disponibili per l'addestramento. È spesso consigliabile iniziare con una grandezza moderata e regolarlo durante il processo di addestramento utilizzando tecniche come la riduzione del tasso di apprendimento o l'utilizzo di ottimizzatori adattativi che aggiustano automaticamente il parametro in base al progresso dell'addestramento. Di conseguenza, è importante trovare un equilibrio ottimale che consenta al modello di apprendere

in modo efficace e convergere a una soluzione di alta qualità (tipo float, valore di default = 1e-3).

- **Early stopping patience:** determina il numero di epoche in cui l'addestramento può continuare senza osservare miglioramenti sulla funzione di loss prima che l'addestramento venga interrotto prematuramente. Durante il processo di addestramento del modello, è comune monitorare la loss sul validation set per valutare la capacità del modello di generalizzare bene ai dati non visti. Quando il valore smette di migliorare per un numero specificato di epoche consecutive (definite come "patience"), l'addestramento viene interrotto anticipatamente per prevenire l'overfitting. Questo significa che il modello viene salvato nella sua migliore configurazione fino a quel momento, evitando un elevato adattamento ai dati di addestramento e consentendo una migliore generalizzazione ai dati futuri (tipo int o none, valore di default = 20).

Per implementare l'ottimizzazione degli iperparametri in AutoGluon, è necessario innanzitutto creare un oggetto `TimeSeriesPredictor`. Questo oggetto richiede alcuni parametri che devono essere specificati durante la sua creazione, ovvero:

- **prediction\_length:** determina l'orizzonte temporale che il modello deve prevedere.
- **freq:** indica la frequenza delle osservazioni della serie temporale.
- **target:** la variabile obiettivo della previsione della serie temporale.
- **path:** il percorso in cui verranno salvati tutti i file relativi al modello.
- **eval\_metric:** metrica da minimizzare durante l'addestramento del modello.

Di seguito è riportato il codice per inizializzare l'oggetto `TimeSeriesPredictor` e successivamente chiamare il metodo `.fit`, fornendo il dataset di addestramento e gli iperparametri da ottimizzare.

```
# Build predictor
predictor = TimeSeriesPredictor(
    prediction_length=12,
    freq='W-MON',
    target='target',
    path='model/TFT-final_HPO',
    eval_metric='MASE'
)

from autogluon.common import space

predictor.fit(
    train_data,
    hyperparameters = {
```



```

        "TemporalFusionTransformer": {
            "hidden_dim": space.Categorical(32, 64, 128),
            "variable_dim": space.Categorical(32, 64, 128),
            "num_heads": space.Categorical(1,2,4),
            "dropout_rate": space.Real(0.1, 0.5),
            "max_epochs": space.Categorical(50, 100, 150),
            "batch_size": space.Categorical(64, 128, 256),
            "lr": space.Categorical(0.1, 0.01, 0.001),
            "early_stopping_patience" : space.Categorical(10, 20)
        },
    },
    hyperparameter_tune_kwargs="auto",
    refit_full=True,
    num_val_windows=5,
    random_seed=42,
    enable_ensemble = False
)

```

Listing 4.2: Creazione oggetto TimeSeriesPredictor e configurazione ottimizzazione iperparametri

Come si evince dal blocco di codice, per alcuni parametri, è stato scelto di mantenere i valori di default, mentre per quelli ritenuti più cruciali è stata eseguita un'ottimizzazione attraverso la valutazione di una gamma di valori. Inoltre, è importante notare che è necessario specificare il parametro *hyperparameter\_tune\_kwargs*. Questo parametro opzionale è utilizzato per definire la strategia di ottimizzazione degli iperparametri (HPO, Hyperparameter Optimization) e per configurare i relativi argomenti per l'ottimizzazione.

Impostandolo su "auto", viene attivata la ricerca bayesiana per i modelli di previsione basati su deep learning, come il Temporal Fusion Transformer. La ricerca bayesiana è un metodo di ottimizzazione degli iperparametri che si basa sulla costruzione di un modello probabilistico delle prestazioni del modello in funzione degli iperparametri. Questo modello viene aggiornato continuamente all'aumentare delle osservazioni, consentendo di focalizzare la ricerca sugli iperparametri più promettenti. In sostanza, cerca di bilanciare l'esplorazione dello spazio degli iperparametri con lo sfruttamento delle informazioni acquisite durante il processo di ottimizzazione, al fine di trovare rapidamente una soluzione vicina all'ottimale con un numero limitato di valutazioni della funzione obiettivo.

Gli iperparametri soggetti a ottimizzazione sono riportati nella Tabella 4.3, insieme ai valori testati per ciascuno di essi.

Iperparametro	Gamma di valori	Valore finale
Hidden dim	32, 64, 128	128
Variable dim	32, 64, 128	32
Num head	1, 2, 4	2
Dropout rate	[0.1; 0.5]	0.158
Max epochs	50, 100, 150	100
Batch size	64, 128, 256	128
Learning rate	0.1, 0.01, 0.001	0.01
Early stopping patience	10, 20	10

Tabella 4.3: Iperparametri ottimizzati

Dopo aver completato l'ottimizzazione degli iperparametri, è stato osservato un significativo miglioramento nelle prestazioni del modello Temporal Fusion Transformer in termini di metrica di valutazione da minimizzare, cioè il Mean Absolute Square Error (MASE). Inizialmente, il modello ha registrato un valore di MASE pari a 0,536, ma grazie all'ottimizzazione degli iperparametri, questa cifra è stata notevolmente ridotta a 0,436. Questo miglioramento del 18,7% è indicativo della capacità del modello di adattarsi meglio ai dati di training e di produrre previsioni più accurate. La riduzione del MASE suggerisce che il modello sia in grado di catturare in modo più efficace i pattern nei dati di serie temporali e di generalizzarli per effettuare previsioni più precise sulle nuove osservazioni. Questo risultato conferma l'importanza dell'ottimizzazione degli iperparametri nel processo di sviluppo dei modelli di previsione delle serie temporali, evidenziando come anche piccoli aggiustamenti possano portare a miglioramenti significativi nelle prestazioni del modello.

## 4.5 Valutazione prestazioni del modello

Nell'ultima fase di implementazione del modello, ci concentriamo sulla valutazione delle prestazioni del Temporal Fusion Transformer ottimizzato facendo uso dei dati di test. È importante sottolineare che questi dati non sono mai stati visti dal modello durante il processo di addestramento e ottimizzazione degli iperparametri. La valutazione verrà eseguita su tutte e cinque le serie temporali relative alle filiali di Italia, Gran Bretagna, Francia, Polonia e Romania.

Questo approccio ci consente di valutare l'efficacia del modello nel generare previsioni accurate su una varietà di contesti e condizioni. Misurando le prestazioni del modello sui dati di test, possiamo ottenere una valutazione obiettiva della sua capacità di generalizzazione e di adattamento ai nuovi dati, fornendo informazioni cruciali per valutare

l'affidabilità delle previsioni nelle diverse situazioni.

Per valutare il modello, è necessario utilizzare il metodo *predict* sull'oggetto `TimeSeriesPredictor` precedentemente creato, al quale vanno passati i dati su cui si desidera effettuare una previsione.

```
# Make prediction on training data
predictions = predictor.predict(train_data, random_seed=42)
```

Listing 4.3: Codice da eseguire per ottenere le previsioni

In questo caso, il modello prevederà i prossimi 12 passi temporali, corrispondenti a 12 settimane a partire dal 30 gennaio 2023 fino al 17 aprile 2023. Successivamente, i valori previsti dal modello verranno confrontati con i dati effettivi del set di test mediante un grafico per verificare la correttezza delle previsioni.

#### 4.5.1 Risultati sperimentazione

Innanzitutto, sono state osservate le previsioni del modello relative alla filiale Italia del gruppo Contoso. È evidente come queste previsioni siano in grado di seguire l'andamento dei dati effettivi, tuttavia, in alcuni casi, risultano leggermente superiori rispetto ai valori dei dati di test, mentre in altri sono di poco inferiori. Questo risultato è visibile nel grafico riportato in figura 4.14; la linea blu rappresenta i valori precedenti alle previsioni, mentre la linea verde si riferisce ai dati effettivi futuri e infine, la linea arancione indica le previsioni generate dal modello.

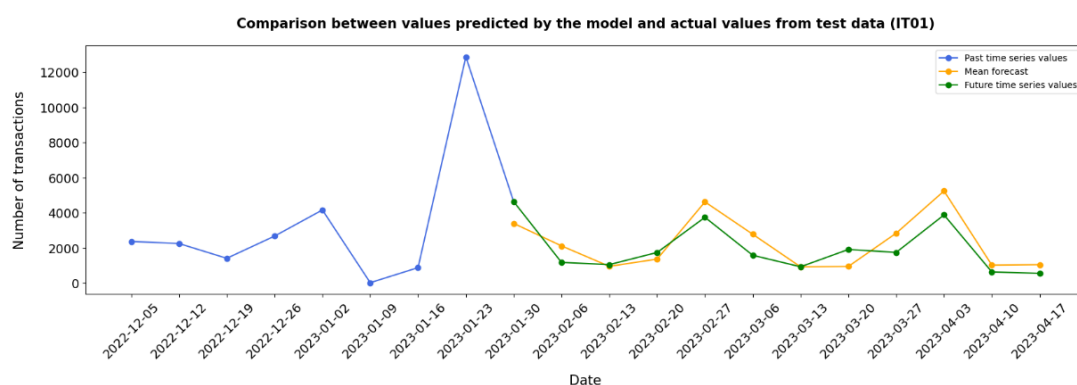


Figura 4.14: Confronto previsioni del modello con i valori effettivi per la filiale Italia

Per quanto riguarda l'azienda Polacca, è evidente come il modello riesca a seguire fedelmente lo stesso andamento dei dati effettivi. In gran parte dei casi, le previsioni coincidono quasi esattamente con i valori reali, eccezion fatta per alcuni passaggi temporali in cui si registra un errore minimo da parte del modello. Anche in questo caso il risultato è visibile nella figura 4.15

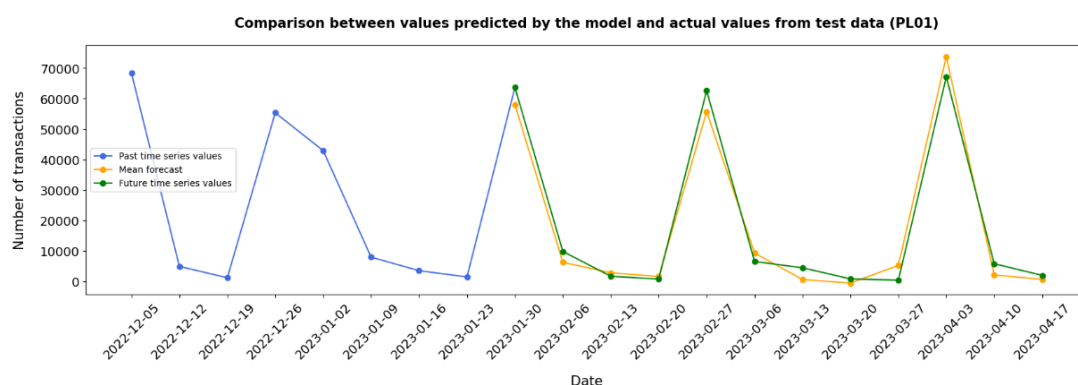


Figura 4.15: Confronto previsioni del modello con i valori effettivi per la filiale Polonia

Per le filiali con sede nel Regno Unito e in Francia, le previsioni mostrano un'aderenza all'andamento dei dati effettivi solo in alcuni segmenti. Questo fenomeno potrebbe essere attribuito a picchi inaspettati causati da ritardi nella chiusura del bilancio. Ad esempio, entrambe le filiali mostrano un errore significativo nella settimana che inizia il 3 aprile 2023. Inoltre, nella settimana tra fine gennaio e inizio febbraio, con inizio il 30 gennaio, il modello commette un errore sostanziale nella previsione, non prevedendo un picco così elevato. Questa discrepanza potrebbe derivare dal fatto che entrambe le aziende abbiano posticipato la chiusura annuale del bilancio 2022, causando un aumento improvviso del volume di transazioni da registrare nel sistema. I grafici relativi sono consultabili nelle figure 4.16 e 4.17.

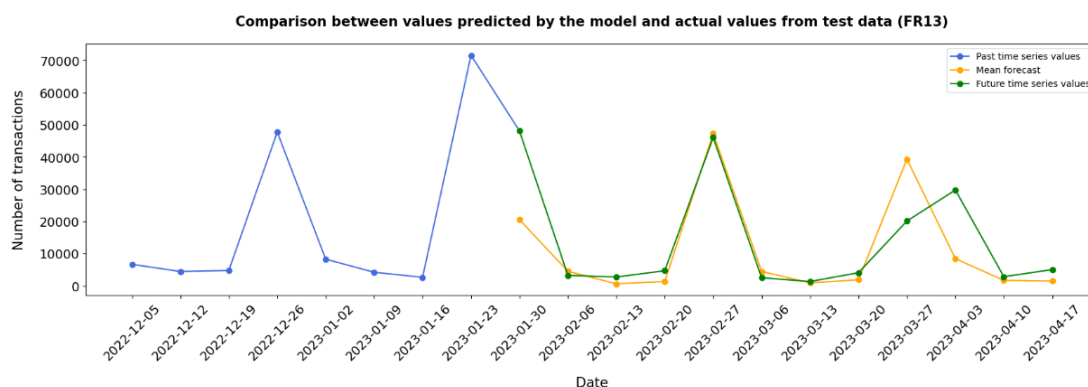


Figura 4.16: Confronto previsioni del modello con i valori effettivi per la filiale Francia

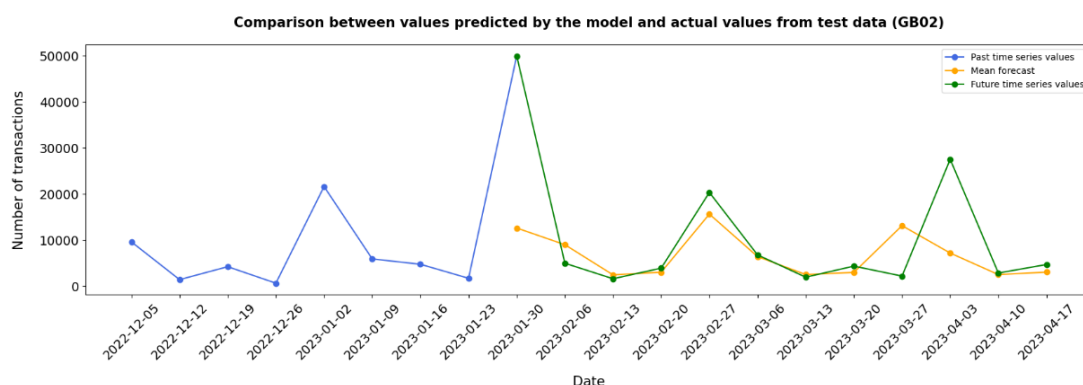


Figura 4.17: Confronto previsioni del modello con i valori effettivi per la filiale Gran Bretagna

Infine, per quanto riguarda la sede situata in Romania, il modello è in grado di seguire quasi sempre il trend del numero di transazioni, ad eccezione di due settimane in cui commette un errore piuttosto evidente. Queste due settimane iniziano rispettivamente il 20 febbraio 2023 e il 6 marzo 2023. Anche in questi casi, gli errori potrebbero essere attribuiti a ritardi nella chiusura contabile del mese, soprattutto nella settimana di marzo. Tuttavia, è proprio in situazioni come queste che il sistema di previsione potrebbe dimostrarsi utile per ottimizzare il processo di chiusura contabile. Il grafico è visibile in figura 4.18.

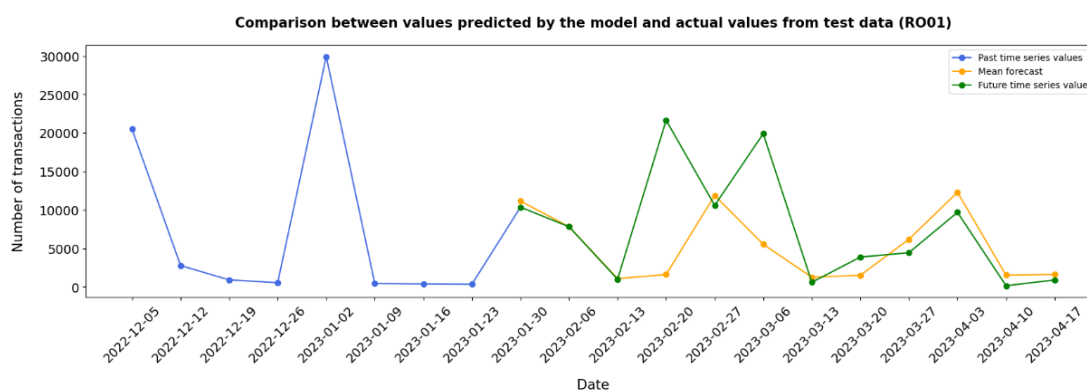


Figura 4.18: Confronto previsioni del modello con i valori effettivi per la filiale Romania

Concludendo, è importante sottolineare che il modello è stato sviluppato considerando tutte le serie temporali relative alle varie filiali prese in esame. Ma questa scelta potrebbe aver influito sulla precisione complessiva del modello. Inizialmente, è stato sperimentato un approccio in cui veniva sviluppato un modello esclusivamente per la sede italiana. Questo esperimento ha prodotto risultati leggermente migliori sia in termini di previsioni, come evidenziato dal grafico visibile in figura 4.19, sia in termini di MASE, con un valore di 0.392, leggermente migliore rispetto al modello addestrato e ottimizzato utilizzando tutte e cinque le serie temporali. Tuttavia, l'approccio di sviluppare un modello per ogni filiale potrebbe risultare poco pratico in termini di costi computazionali, specialmente

per aziende con numerose sedi. Pertanto, è stata preferita l'opzione di utilizzare un unico modello che comprendesse tutte le filiali, nonostante alcuni compromessi nella precisione.

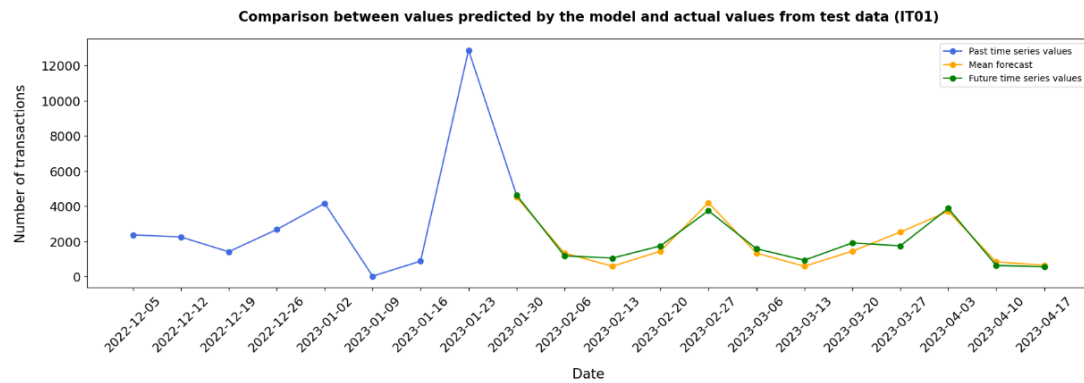


Figura 4.19: Risultati del modello sviluppato per la singola sede Italiana

## Capitolo 5

# Ingegnerizzazione della soluzione

In questo capitolo, viene discussa la fase finale del progetto, che consiste nella realizzazione di un microservizio altamente efficiente ed efficace in grado di soddisfare gli obiettivi e i vincoli specifici del progetto. L'obiettivo principale è sviluppare un prodotto finale che sia in grado di prevedere il numero di transazioni per le 12 settimane successive all'ultima osservazione della serie temporale fornita al modello. Questo processo avviene quando viene richiamato il metodo `'predict'` sull'oggetto `'TimeSeriesPredictor'`.

Per raggiungere questo obiettivo, è necessario progettare e implementare un sistema che sia in grado di gestire in modo efficiente le richieste di previsione da parte degli utenti, garantendo al contempo un alto livello di precisione e tempi di risposta rapidi. Ciò implica la progettazione di un'architettura scalabile e resiliente, in grado di gestire un elevato volume di richieste in modo affidabile e senza compromettere le prestazioni del sistema.

Inoltre, è fondamentale distribuire il modello all'interno di un'architettura cloud per garantire l'accessibilità delle previsioni da qualsiasi sistema tramite richieste HTTP. Questo approccio consente una maggiore flessibilità nell'uso del servizio di previsione, consentendo agli utenti di accedere alle predizioni in modo rapido e conveniente tramite una semplice API.

L'adozione di un'architettura cloud offre numerosi vantaggi, tra cui la scalabilità orizzontale per gestire un elevato volume di richieste, l'affidabilità dei servizi, la possibilità di ridimensionare le risorse in base alle esigenze del carico di lavoro e la facilità di gestione e manutenzione del sistema. La distribuzione del modello all'interno di un ambiente cloud consente una maggiore flessibilità nell'integrazione con altri servizi e applicazioni aziendali, facilitando l'adozione e l'integrazione del sistema di previsione all'interno dell'ecosistema IT aziendale.

Verrà descritta l'architettura progettata per implementare il servizio di previsione e la fase di sviluppo associata. Saranno illustrati tutti i passaggi necessari per ottenere previsioni accurate attraverso un endpoint accessibile tramite richieste HTTP. In primo luogo verrà esposta l'architettura ad alto livello, delineando i componenti principali e le loro interazioni. Successivamente, verrà approfondita la fase di sviluppo, illustrando

i dettagli tecnici e pratici relativi all'implementazione del servizio, compresi gli script Python necessari.

## 5.1 Architettura proposta

L'architettura realizzata per il servizio di previsione delle serie temporali comprende una serie di passaggi chiave, elencati di seguito:

1. **Ottenimento dei dati:** per ottenere i dati necessari per eseguire il processo di inferenza, viene utilizzato uno script Python che stabilisce una connessione con il database salvato in locale. Attraverso una query SQL, vengono recuperate le transazioni di chiusura contabile, che costituiscono i dati storici utilizzati per l'addestramento del modello.
2. **Registrazione del modello:** dopo aver recuperato i dati più recenti e aver attuato la fase di preprocessing spiegata in sezione 4.2, essi vengono usati per addestrare il modello di previsione delle serie temporali precedentemente sviluppato. Il modello addestrato viene quindi salvato all'interno dell'area di lavoro di Azure Machine Learning Studio, preparandolo per la distribuzione successiva.
3. **Creazione servizio:** durante questa fase, viene creato il servizio utilizzando uno script Python, il quale rende il servizio accessibile attraverso un URI e una chiave. Inoltre, viene creata un'immagine di un ambiente Python necessaria per effettuare l'inferenza. Maggiori dettagli su questa fase verranno forniti nelle sezioni successive.
4. **Utilizzo del servizio:** racchiude lo script Python per accedere al servizio, in modo da ottenere il numero di transazioni di chiusura contabile previsto per le prossime 12 osservazioni temporali.

Nella Figura 5.1 è rappresentato un diagramma che illustra il funzionamento dell'architettura e i suoi componenti correlati.



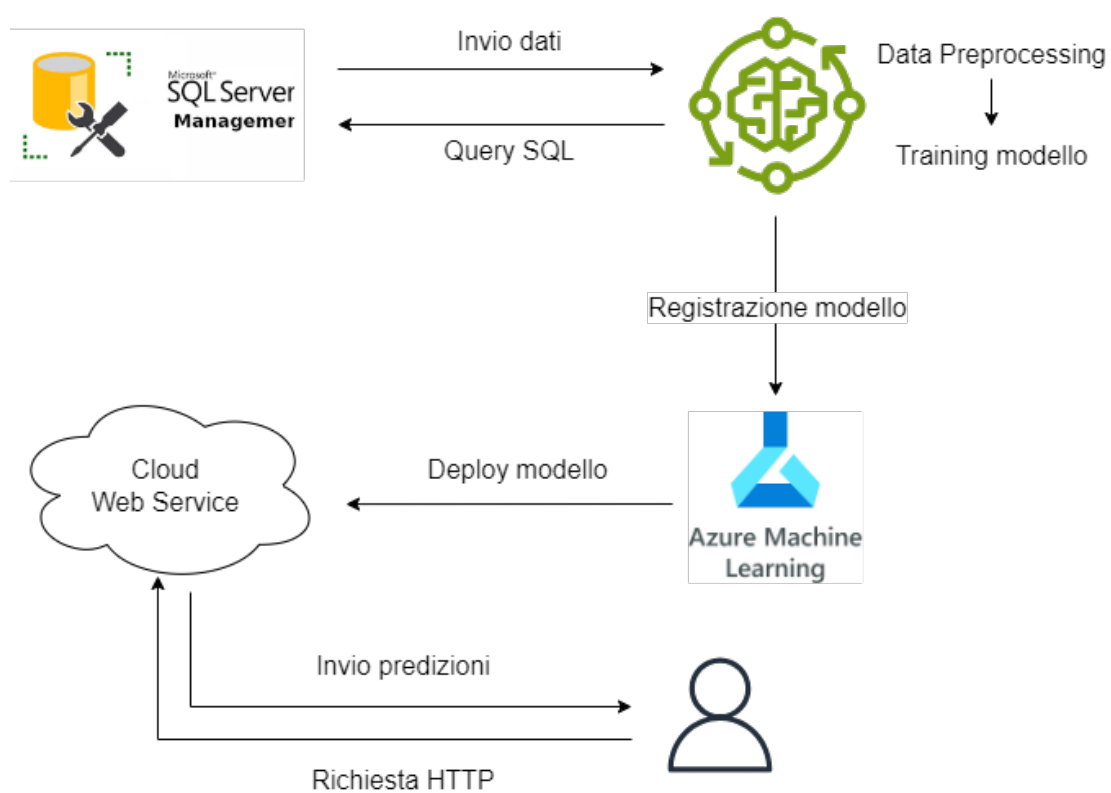


Figura 5.1: Componenti e funzionamento dell'architettura del servizio

## 5.2 Ottenimento dati

Come precedentemente spiegato, per acquisire i dati più recenti dal database, è necessario eseguire uno script Python. Questo script stabilisce una connessione locale al database e successivamente esegue una query SQL per recuperare le informazioni desiderate. Per eseguire lo script, è necessario importare *pyodbc*, un modulo open source che rende semplice l'accesso ai database ODBC (Open Database Connectivity).

```
# Connessione al database utilizzando l'autenticazione Windows
conn = pyodbc.connect('DRIVER={SQL Server};
    SERVER=*****;DATABASE=****;Trusted_Connection=yes;')

cursor = conn.cursor()

# Esecuzione della query
query = """
SELECT distinct
    GJAE.RECID as GjaeRecId,
    GJE.CreatedDateTime,
    GJE.SubledgerVoucherDataAreaId,
    JC.Label AS JournalCategory,
    PT.Label AS PostingTypeLabel,
    GJAE.PostingType AS PostingTypeValue,
    CPA.Description as ClosingActivityDescription,
    CPA.Id as ClosingActivityId
FROM GENERALJOURNALENTTRY GJE
JOIN GENERALJOURNALACCOUNTENTRY GJAE ON GJE.RECID =
    GJAE.GeneralJournalEntry
JOIN POSTINGTYPE PT ON PT.Value = GJAE.POSTINGTYPE
JOIN JOURNALCATEGORY JC ON GJE.JOURNALCATEGORY = JC.Value
JOIN DIMENSIONATTRIBUTEVALUECOMBINATION DAVC ON GJAE.LEDGERDIMENSION =
    DAVC.RECID
JOIN DIMATTRIBUTEGROUPCHARTOFACCOUNTS DAGCA ON DAGCA.VALUE =
    DAVC.GROUPCHARTOFACCOUNTSVALUE
left JOIN LEDGERJOURNALTRANSACTION LJT ON LJT.GENERALJOURNALENTTRY =
    GJE.RECID
left JOIN LEDGERJOURNALTYPE LJTY ON LJTY.Value = LJT.JOURNALTYPE
join MAPPINGCLOSINGPERIODACTIVITY MCPA ON MCPA.postingtypelabel =
    pt.label and mcpa.journalcategorylabel = jc.label
join CLOSINGPERIODACTIVITY CPA ON CPA.Id = MCPA.ActivityId
WHERE
    (GJE.SUBLEDGERVOUCHERDATAAREAID IN ('it01', 'GB02', 'fr13', 'PL01',
        'ro01'))
    AND (GJE.CreatedDateTime >= '2020-01-01')
ORDER BY
    GJE.CREATEDDATETIME;
"""
```

Listing 5.1: Connessione al database e query SQL per l'ottenimento dei dati

Una volta completata la query, i dati vengono salvati in formato CSV per un successivo utilizzo.

```
# Esecuzione della query e salvataggio dei risultati in un DataFrame di
# pandas
df = pd.read_sql(query, conn)

# Salvataggio del DataFrame in un file CSV
df.to_csv('test.csv', index=False)

# Chiusura della connessione
cursor.close()
conn.close()
```

Listing 5.2: Codice da eseguire per ottenere le predizioni

Questo processo è necessario ogni qual volta si vogliano ottenere delle nuove previsioni basate sull'ultima data di osservazione

### 5.3 Registrazione del modello

Per registrare un modello all'interno della piattaforma Azure Machine Learning Studio dopo aver eseguito l'addestramento, è necessario utilizzare l'oggetto Experiment. Questo è un componente cruciale nell'ecosistema di sviluppo di modelli di machine learning. Esso rappresenta il punto di ingresso principale per organizzare, tracciare e gestire le iterazioni dei modelli durante il processo di sviluppo e sperimentazione. In altre parole, l'Experiment è una sorta di contenitore virtuale che consente di monitorare e gestire il flusso di lavoro relativo alla creazione e alla valutazione dei modelli. Per creare un nuovo esperimento è necessario specificare l'area di lavoro e un nome significativo che rifletta l'obiettivo o il contesto del progetto. Una volta creato, questo esperimento fungerà da contenitore per tutte le versioni di modelli che verranno addestrate e valutate durante il processo di sviluppo; ogni qual volta che verrà eseguito un nuovo addestramento del modello, verrà inizializzata l'esecuzione dell'esperimento, come è visibile nel codice seguente.

```
from azureml.core import Experiment , Model , Dataset

# Create an Azure ML experiment in your workspace
experiment = Experiment(workspace = ws, name = "Train-TFT-Model")
run = experiment.start_logging(snapshot_directory=None)
print("Starting experiment:", experiment.name)
```

Listing 5.3: Codice da eseguire per ottenere le predizioni

Una volta avviata un'esecuzione dell'esperimento, questo terrà traccia di tutte le informazioni rilevanti, come i parametri del modello, le metriche di valutazione, i log di output e altro ancora. Una volta terminato l'addestramento viene utilizzata una funzione che carica una cartella contenente il modello sull'archivio associato all'esperimento corrente. Successivamente, tramite la funzione `register_model`, il modello viene registrato nell'ambiente di Azure Machine Learning Studio.

```
# Upload model
run.upload_folder(name='outputs/TFT-Final', path='model/TFT-Final')

# Register BEST model
run.register_model(model_path='outputs/TFT-Final',
                    model_name='TFT-Final',
                    tags={'Training context': 'Inline Training'})

print('Model: TFT-Final trained and registered.')
```

Listing 5.4: Codice da eseguire per ottenere le predizioni

Una volta eseguito questo codice, il modello verrà caricato nell'archivio di Azure Machine Learning Studio e registrato come un oggetto nell'ambiente, rendendolo disponibile per l'uso in altre parti del progetto o per la distribuzione come servizio di inferenza.

## 5.4 Creazione servizio

In questa sezione, viene illustrata la fase chiave relativa all'ingegnerizzazione del servizio software, che rappresenta uno degli obiettivi principali del progetto di tesi. È importante notare che esistono tre diverse modalità per distribuire un servizio web all'interno di Azure Machine Learning Studio [25]:

- **LocalWebService:** consente di distribuire un modello in locale, eseguendolo direttamente sul proprio ambiente di sviluppo o su una macchina locale. Questo è utile per lo sviluppo e il testing in ambienti controllati.
- **AciWebService:** permette di distribuire un modello su Azure Container Instances (ACI), che è un servizio di calcolo serverless, scalabile e a bassa latenza. È ideale per carichi di lavoro leggeri o per scopi di sviluppo e testing.
- **AksWebserveice:** la creazione del servizio si appoggia su Azure Kubernetes Service (AKS), un servizio di orchestratore di contenitori completamente gestito e altamente scalabile. Questa soluzione è ideale per carichi di lavoro più pesanti e per applicazioni di produzione che necessitano di una disponibilità elevata e di una scalabilità dinamica.

In questo contesto, è stata effettuata la scelta di utilizzare un servizio di tipo ACI. Prima di procedere con la distribuzione del servizio, è fondamentale definire due componenti cruciali: lo script Python utilizzato per fare inferenza, comunemente denominato `score.py`, e un'immagine dell'ambiente virtuale che ospiterà l'esecuzione dello script.

Lo script Python contiene due importanti funzioni:

- **init()**: questa funzione viene richiamata quando il contenitore viene inizializzato o avviato, tipicamente dopo la creazione o l'aggiornamento della distribuzione. Qui è possibile scrivere la logica necessaria per eseguire operazioni di inizializzazione, come ad esempio memorizzare nella memoria cache il modello.
- **run(data\_raw)**: questa funzione viene richiamata ogni volta che l'utente richiede l'accesso all'endpoint tramite una richiesta HTTP per ottenere le predizioni. Per effettuare la richiesta è necessario fornire i dati nel corpo della richiesta in formato JSON<sup>7</sup>. Una volta che il servizio riceve i dati, esegue la predizioni sul modello precedentemente memorizzato con la funzione `init()`, e dopo aver eseguito alcune operazioni di formattazione sulle predizioni, restituisce i risultati in formato JSON. Vedremo i dettagli di come usufruire del servizio nella sezione successiva.

Inoltre, è fondamentale definire le librerie da importare essenziali per l'esecuzione del codice. Di seguito viene mostrato il codice sviluppato per implementare le due funzioni precedentemente nominate.

```
def init():  
  
    # Define model variable  
    global model  
  
    # Get the path to the deployed model file and load it  
    model_path = Model.get_model_path('TFT-Final') # Forecasting_model  
    model = TimeSeriesPredictor.load(model_path)
```

Listing 5.5: Codice implementazione funzione `init()`

```
def run(data_raw):  
  
    # Convert json dat to pandas df  
    data = pd.read_json(data_raw)  
  
    predictions = model.predict(data)  
  
    # Extract 'item_id' and 'timestamp' from the index and add them as  
    columns
```

<sup>7</sup>JavaScript Object Notation, formato per lo scambio dati basato sul linguaggio di programmazione JavaScript.

```
predictions['item_id'] =
    predictions.index.get_level_values('item_id')
predictions['timestamp'] =
    predictions.index.get_level_values('timestamp').strftime('%Y-%m-%d')

# Select only the columns 'item_id', 'timestamp', and 'mean'
predictions_selected = predictions[['item_id', 'timestamp', 'mean']]

# Convert the selected DataFrame into JSON format
predictions_json = predictions_selected.to_json(orient='records')

return predictions_json
```

Listing 5.6: Codice implementazione funzione run()

Riguardo all'immagine dell'ambiente virtuale su cui verrà eseguito lo script di inferenza, questo svolge un ruolo fondamentale nel garantire che il modello possa essere eseguito in un ambiente coerente e affidabile. È necessario specificare sia la versione di Python da utilizzare, sia le dipendenze e i pacchetti utili per il corretto funzionamento del modello. Questo include sia le librerie di base come NumPy e Pandas, la libreria AutoGluon. Una volta definita la configurazione dell'ambiente virtuale, questa viene salvata come un file YAML<sup>8</sup>, garantendo coerenza e facilitando la gestione delle dipendenze del progetto.

```
## inference config
from azureml.core import Environment
from azureml.core.model import InferenceConfig
from azureml.core.conda_dependencies import CondaDependencies

# Add the dependencies for our model (AzureML defaults is already
    included)
myenv = CondaDependencies()

myenv.set_python_version('3.10')

# Add conda packages
myenv.add_conda_package('numpy')
myenv.add_conda_package('pip')
myenv.add_conda_package('pandas')

# Add pip packages
myenv.add_pip_package('autogluon')

# Save the environment config as a .yaml file
env_file_name = "TFT_env.yaml"
```

<sup>8</sup> Linguaggio leggibile in chiaro per la serializzazione di dati che viene spesso impiegato per la scrittura dei file di configurazione.

```
env_file = os.path.join(experiment_folder, env_file_name)

with open(env_file, "w") as f:
    f.write(myenv.serialize_to_string())
print("Saved dependency info in", env_file)
```

Listing 5.7: Codice per la definizione dell'immagine dell'ambiente Python

Assicurarsi che l'ambiente sia adeguatamente configurato e che tutte le dipendenze siano correttamente installate è essenziale per garantire che il servizio di inferenza funzioni in modo ottimale e produca previsioni accurate e affidabili.

Per completare la creazione del servizio web su Azure Container Instance, è necessario inizializzare la classe `InferenceConfig`, fornendo lo script `score.py` e il file di configurazione YAML. Successivamente, definiamo la modalità di deployment specificando le risorse di calcolo desiderate, come il numero di core della CPU e la quantità di memoria da allocare per il servizio. Infine, viene richiamato il metodo `deploy` sull'oggetto `Model`, caricando il modello e passando tutte le impostazioni precedentemente configurate. Una volta completato questo processo, il servizio sarà pronto per l'uso nell'inferenza delle previsioni.

```
from azureml.core.webservice import AciWebservice
from azureml.core.model import InferenceConfig

# Configure the scoring environment
inference_config = InferenceConfig(runtime="python",
    entry_script=script_file, conda_file=env_file)

# Deployment configuration with HTTPS enabled
deployment_config = AciWebservice.deploy_configuration(cpu_cores=1,
    memory_gb=1, auth_enabled=True)

# Service deployment
service_name = "forecastingprediction"
model = ws.models['TFT-Final']

# Deploy the model as a web service
service = Model.deploy(workspace=ws,
    name=service_name,
    models=[model],
    inference_config=inference_config,
    deployment_config=deployment_config)

# Wait for deployment to complete
service.wait_for_deployment(True)
```

Listing 5.8: Codice per la distribuzione del modello





```
# HTTP request
predictions = requests.post(uri, df_json, headers = headers,
                             timeout=None)
```

Listing 5.9: Codice per effettuare la richiesta HTTP e ottenere le predizioni

Dopo aver ricevuto la risposta dal servizio, è possibile procedere con la visualizzazione dei risultati attraverso un grafico. Questo rappresenta i passi temporali precedenti alle predizioni insieme alle previsioni stesse, uno per ogni filiale. L'obiettivo è fornire una rappresentazione visiva dei dati storici insieme alle previsioni future, consentendo una rapida comprensione dei trend e delle eventuali variazioni nel tempo. Tale visualizzazione è utile per l'analisi dei risultati e per definire il carico di lavoro necessario per chiudere il bilancio mensile.

Di seguito sono presentati i grafici relativi alle previsioni del numero di transazioni per il periodo compreso tra il 1° maggio 2023 e il 17 luglio 2023, considerando tutte le diverse filiali prese in esame.

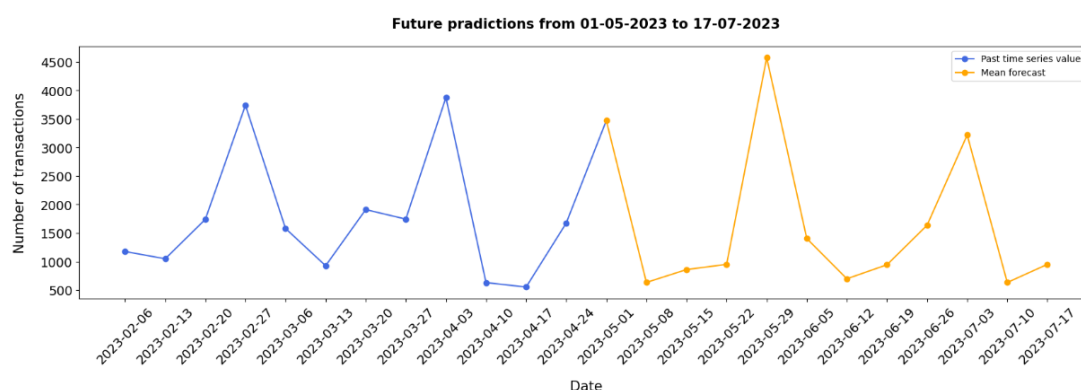


Figura 5.3: Previsioni numero di transazioni da registrare per la filiale Italia

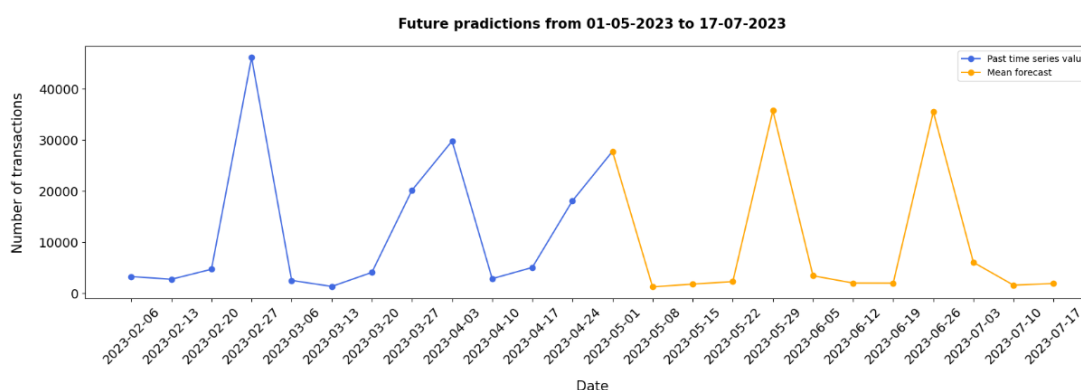


Figura 5.4: Previsioni numero di transazioni da registrare per la filiale Francia

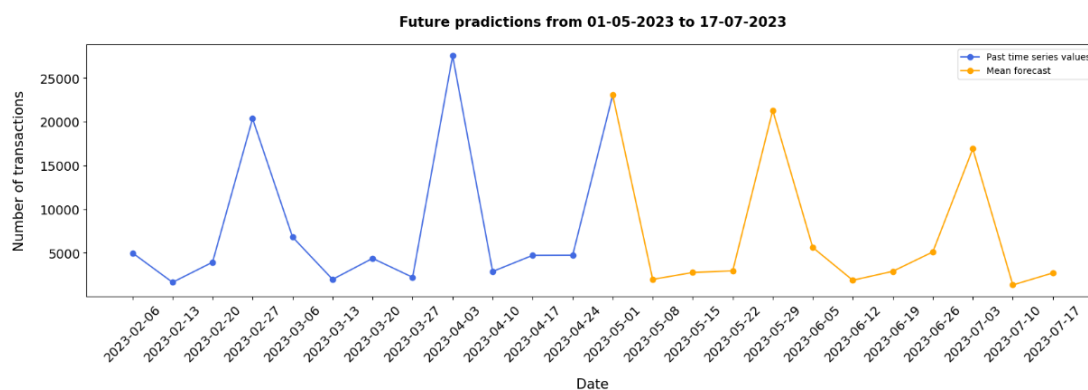


Figura 5.5: Previsioni numero di transazioni da registrare per la filiale Gran Bretagna

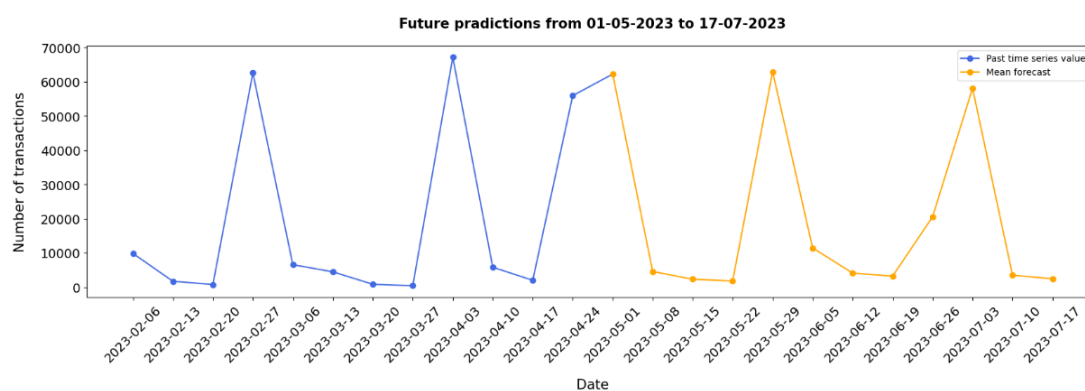


Figura 5.6: Previsioni numero di transazioni da registrare per la filiale Polonia

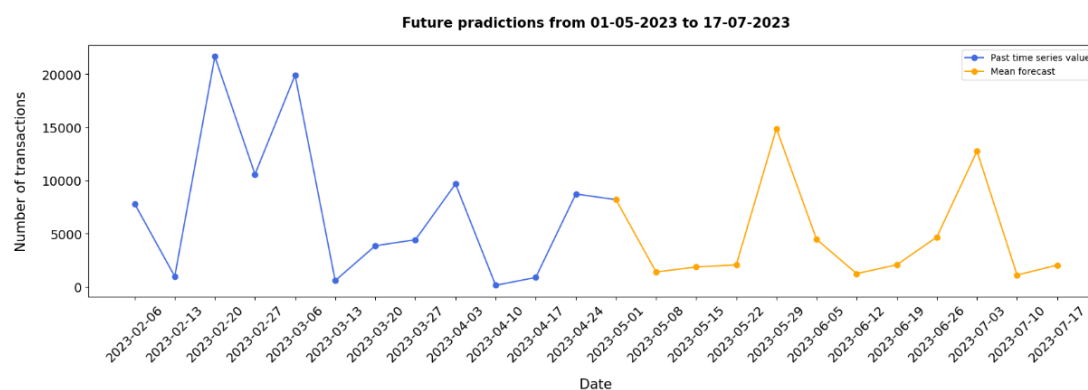


Figura 5.7: Previsioni numero di transazioni da registrare per la filiale Romania

## Capitolo 6

# Conclusioni e sviluppi futuri

In questo capitolo, saranno esaminati gli obiettivi raggiunti durante lo svolgimento della tesi, successivamente saranno affrontati i vari limiti del progetto e i potenziali sviluppi futuri per migliorare e ampliare le funzionalità del sistema realizzato.

### 6.1 Conclusioni

In conclusione, il lavoro di tesi ha raggiunto gli obiettivi prefissati, presentando un servizio software progettato per ottimizzare il processo di chiusura contabile aziendale. Tale sistema si distingue per la sua capacità di predire con una buona precisione il numero di transazioni da registrare, tenendo conto delle specificità di ogni filiale all'interno di una grande azienda. La soluzione proposta si presenta come un'interfaccia accessibile e integrabile in diversi ambienti operativi, consentendo un'interazione fluida attraverso semplici richieste HTTP di tipo post.

Nel corso dello sviluppo del sistema, sono state dettagliatamente illustrate sia la fase di ricerca che la fase di ingegnerizzazione del prodotto. Durante la fase di ricerca, è stata dedicata particolare attenzione allo studio del contesto relativo alla chiusura contabile aziendale, nonché alle diverse metodologie procedurali per l'analisi e la gestione delle previsioni di serie temporali.

Nonostante le varie operazioni di pre-elaborazione abbiano contribuito a migliorare i risultati, l'aggiunta di un maggior numero di osservazioni provenienti non solo dalle sedi prese in considerazione, ma anche da altre filiali, potrebbe sensibilmente migliorare l'accuratezza delle predizioni.

È importante sottolineare che lo sviluppo del servizio è avvenuto senza un coinvolgimento diretto dell'azienda fornitrice dei dati. L'assenza di questa collaborazione ha limitato la capacità di analizzare appieno i requisiti specifici del processo di chiusura contabile aziendale e di ottenere un feedback diretto dagli attori coinvolti nel processo. Pertanto, vi è il potenziale per migliorare ulteriormente il sistema coinvolgendo direttamente i dipendenti e i responsabili del processo. Questo potrebbe consentire una migliore com-

prensione delle esigenze aziendali e portare a previsioni ancora più accurate, in linea con le specifiche necessità e dinamiche operative dell'azienda.

L'obiettivo principale dell'attività di ricerca e sviluppo era valutare la fattibilità di un servizio di predizione per ottimizzare il processo di chiusura contabile aziendale e presentare un prototipo funzionante basato su un caso di studio iniziale. Il lavoro di tesi ha dimostrato chiaramente che è possibile realizzare un tale servizio utilizzando gli strumenti disponibili su Azure, aprendo la strada a future implementazioni e applicazioni in contesti aziendali reali.

Infine, va sottolineato che il servizio sviluppato può essere facilmente personalizzato e adattato alle esigenze di qualsiasi azienda che abbia necessità di rispettare tempistiche rigorose per la chiusura del bilancio. La sua architettura modulare lo rende economicamente sostenibile e altamente scalabile, consentendo un'implementazione efficiente e flessibile in diversi contesti aziendali.

## 6.2 Limiti e sviluppi futuri

Uno dei principali limiti del sistema è che il modello di forecasting implementato non è in grado di prevedere il numero di transazioni da registrare per ogni singola attività di chiusura contabile, ma fornisce solo un valore complessivo unendo tutte le varie attività. Questa limitazione potrebbe influenzare la pianificazione delle risorse umane, poiché una previsione accurata delle transazioni per ogni singola attività consentirebbe una distribuzione ottimale del personale. Una soluzione tentata è stata quella di sviluppare un modello per prevedere il numero di transazioni per ogni specifica attività e filiale. Ma questo approccio non ha prodotto risultati soddisfacenti per la maggior parte delle serie temporali analizzate. Una potenziale alternativa potrebbe consistere nello sviluppo di un modello più specifico e dettagliato, che operi a un livello più basso rispetto all'approccio utilizzato con la libreria AutoGluon. Il nuovo modello potrebbe essere concepito per predire non solo il numero totale di transazioni da registrare nel sistema, ma anche il conteggio distintivo per ciascuna attività [26]. L'adozione di un approccio più personalizzato potrebbe consentire una migliore adattabilità ai diversi contesti aziendali e alle peculiarità di ciascuna attività di chiusura contabile.

D'altra parte, va considerato che l'implementazione di un modello di questo tipo potrebbe richiedere un maggior grado di complessità nell'analisi dei dati e nello sviluppo del modello stesso. È fondamentale valutare attentamente i trade-off tra la complessità del modello e la sua capacità predittiva, bilanciando l'accuratezza delle previsioni con la praticità e l'efficienza nell'uso del sistema.

Un ulteriore problema è quello relativo al data drift di serie temporali [27]; questo si verifica quando le caratteristiche dei dati cambiano nel tempo, rendendo obsolete le relazioni e i modelli sviluppati in precedenza. Questo fenomeno può essere causato da

una serie di fattori, come cambiamenti nei comportamenti dei clienti, nelle condizioni di mercato, nei processi aziendali o nelle variabili esterne.

Il data drift può avere conseguenze negative sulle previsioni, portando a decisioni errate o inefficaci. Per affrontare il suddetto problema, è essenziale implementare un sistema di monitoraggio continuo dei dati e di allerta tempestiva. Questo sistema deve essere in grado di rilevare i cambiamenti nei dati in tempo reale e avvisare gli utenti quando si verificano alterazioni significative. L'identificazione precoce di questi mutamenti nei dati consente di adattare rapidamente i modelli e le strategie di previsione per riflettere le nuove condizioni e mantenere l'accuratezza delle previsioni.

Infatti, un aspetto cruciale nell'ambito del forecasting è la necessità per l'analista di valutare costantemente le prestazioni del modello e assicurarsi che le previsioni siano accurate. Nel caso in cui il modello produca valori errati, è essenziale individuare le cause di tali discrepanze e apportare le modifiche necessarie per migliorare la precisione del modello.

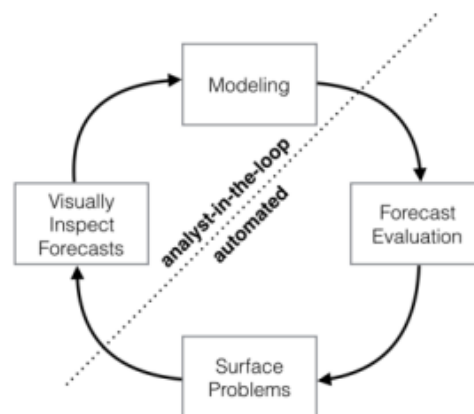


Figura 6.1: Processo di valutazione e miglioramento del modello

# Bibliografia

- [1] Jason Brownlee. How to decompose time series data into trend and seasonality, Dec 2020.
- [2] D. Dickey and Wayne Fuller. Distribution of the estimators for autoregressive time series with a unit root. *JASA. Journal of the American Statistical Association*, 74, 06 1979.
- [3] Selva Prabhakaran. Augmented dickey-fuller (adf) test - must read guide - ml+, Apr 2022.
- [4] Rob Hyndman and G. Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Australia, 3rd edition, 2021.
- [5] Spyros Makridakis, Evangelos Spiliotis, Vassilios Assimakopoulos, Artemios-Anargyros Semenoglou, Gary Mulder, and Konstantinos Nikolopoulos. Statistical, machine learning and deep learning forecasting methods: Comparisons and ways forward. *Journal of the Operational Research Society*, 74(3):840–859, 2023.
- [6] Jocelyn Barker. Machine learning in m4: What makes a good unstructured model? *International Journal of Forecasting*, 2020. M4 Competition.
- [7] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1), 2020. M4 Competition.
- [8] Kasun Bandara, Christoph Bergmeir, and Slawek Smyl. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, 140, 2020.
- [9] Sven F. Crone, Michèle Hibon, and Konstantinos Nikolopoulos. Advances in forecasting with neural networks? empirical evidence from the nn3 competition on time series prediction. *International Journal of Forecasting*, 27(3), 2011. Special Section 1: Forecasting with Artificial Neural Networks and Computational Intelligence  
Special Section 2: Tourism Forecasting.

- 
- [10] Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.
  - [11] Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 2006.
  - [12] Shchur Oleksandr, Turkmen Caner, Erickson Nick, Shen Huibin, Shirkov Alexander, Hu Tony, and Wang Yuyang. Autogluon-timeseries: Automl for probabilistic time series forecasting. In *International Conference on Automated Machine Learning*, 2023.
  - [13] Wes Mckinney. pandas: a foundational python library for data analysis and statistics. *Python High Performance Science Computer*, 01 2011.
  - [14] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3), 2007.
  - [15] Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.
  - [16] Azure machine learning - ml as a service. <https://azure.microsoft.com/en-us/products/machine-learning>.
  - [17] Microsoft sql server management studio. <https://learn.microsoft.com/it-it/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>.
  - [18] Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 2006.
  - [19] Cross-validation strategies for time-series data. [https://quantile.app/blog/cross\\_validation](https://quantile.app/blog/cross_validation).
  - [20] R.G. Brown. *Statistical Forecasting for Inventory Control*. McGraw-Hill, 1959.
  - [21] G.E.P. Box and G.M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day series in time series analysis and digital processing. Holden-Day, 1970.
  - [22] Non-parametric time series (npts) algorithm. <https://docs.aws.amazon.com/forecast/latest/dg/aws-forecast-recipe-npts.html>.
  - [23] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3), 2020.

- 
- [24] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers, 2023.
  - [25] Deploy a web service from registered model(s). [https://azure.github.io/azureml-sdk-for-r/reference/deploy\\_model.html](https://azure.github.io/azureml-sdk-for-r/reference/deploy_model.html).
  - [26] Demand forecasting with the temporal fusion transformer. <https://pytorch-forecasting.readthedocs.io/en/stable/tutorials/stallion.html>.
  - [27] Addressing drifts in time-series forecasting. <https://deepchecks.com/addressing-drifts-in-time-series-forecasting>.