Simone Zambetti

Final project work

13 aprile 2021

Agenda

- Project overview & obiettivi
- Big data analytics:
  - Background
  - Architettura
  - Visualizzazione – PowerBI & Data model
  - Row level access
- Advanced analytics (time series):
  - Background
  - Architettura
  - Risultati

Project overview

Obbiettivi

Big data analytics

Target:

- Cruscotto per il monitoraggio CPM per il business “Eletrification” in ABB
- Ordinato e fatturato non erano tracciati a livello globale; nonostante ci fossero due diversi sistemi che aggregavano i dati da più di 40 ERP, non c'era un quadro completo. L'obiettivo della dashboard è di aggregare questi due sistemi combinando e armonizzando i diversi campi e riconciliandoli con il bilancio ufficiale.
- Migrazione fonte dati a warehouse aziendale in Snowflake, sostituendo una vecchia architettura che si affidava interamente a powerbi dataflows per l'ETL e l'estrazione tramite excel.

Advanced analitycs

- Creare un POC per forecast per l’ordinato di una singola product line usando deep learning e un modello lineare più semplice
- Connettere Databricks a snowflake e azure blob storage
- Visualizzazione del forecast per i prossimi sei mesi

Big data analytics

Dashboard

Migrazione

Da dataflows a snowflake

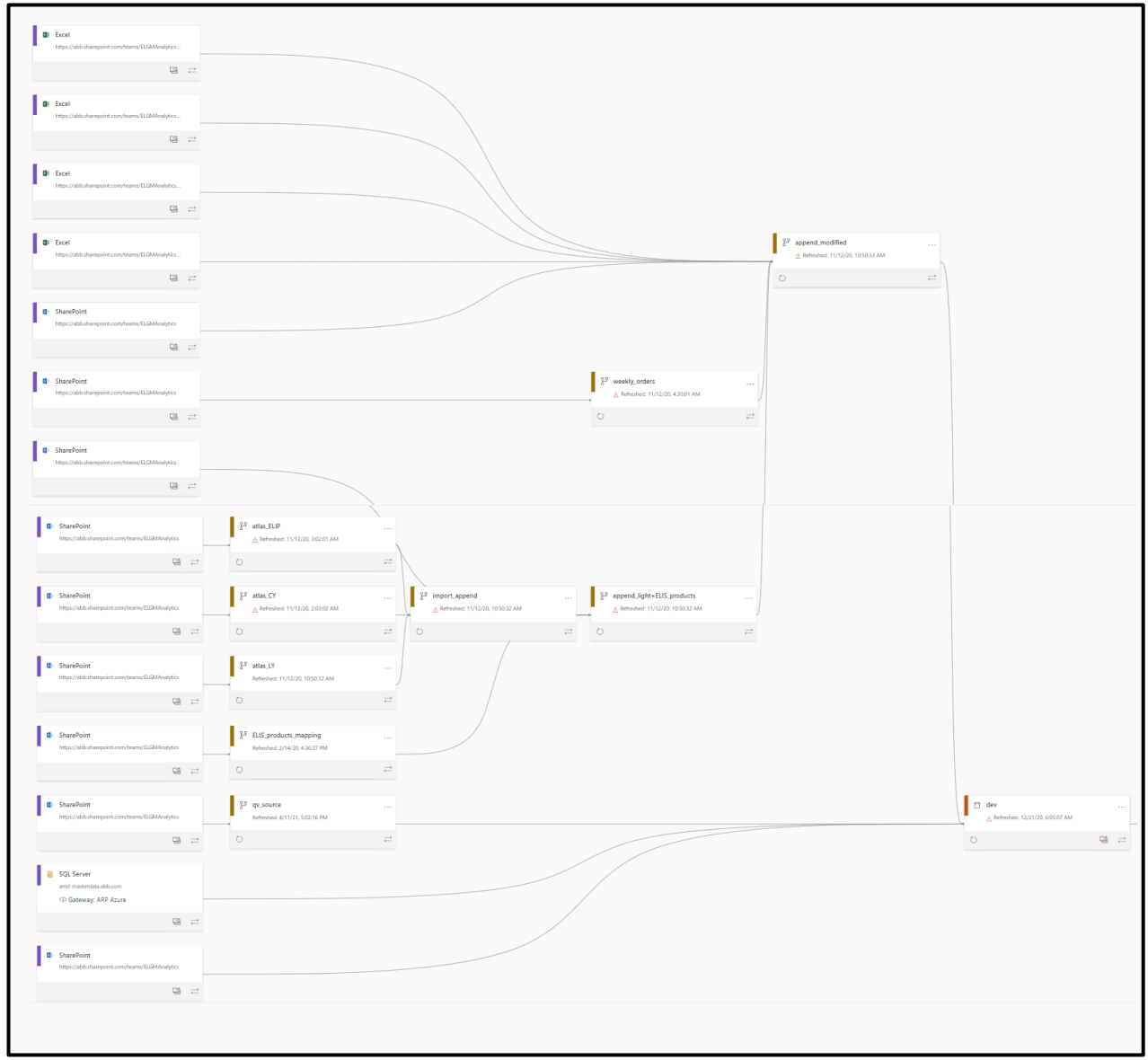
Background del progetto

In precedenza, vi era già una dashboard in produzione, la cui architettura si affidava a powerbi dataflows come transformation and load (a destra), inoltre che ad una workstation per effettuare estrazione tramite macro in excel. Questa architettura aveva molte lacune, quindi ho proposto una migrazione.

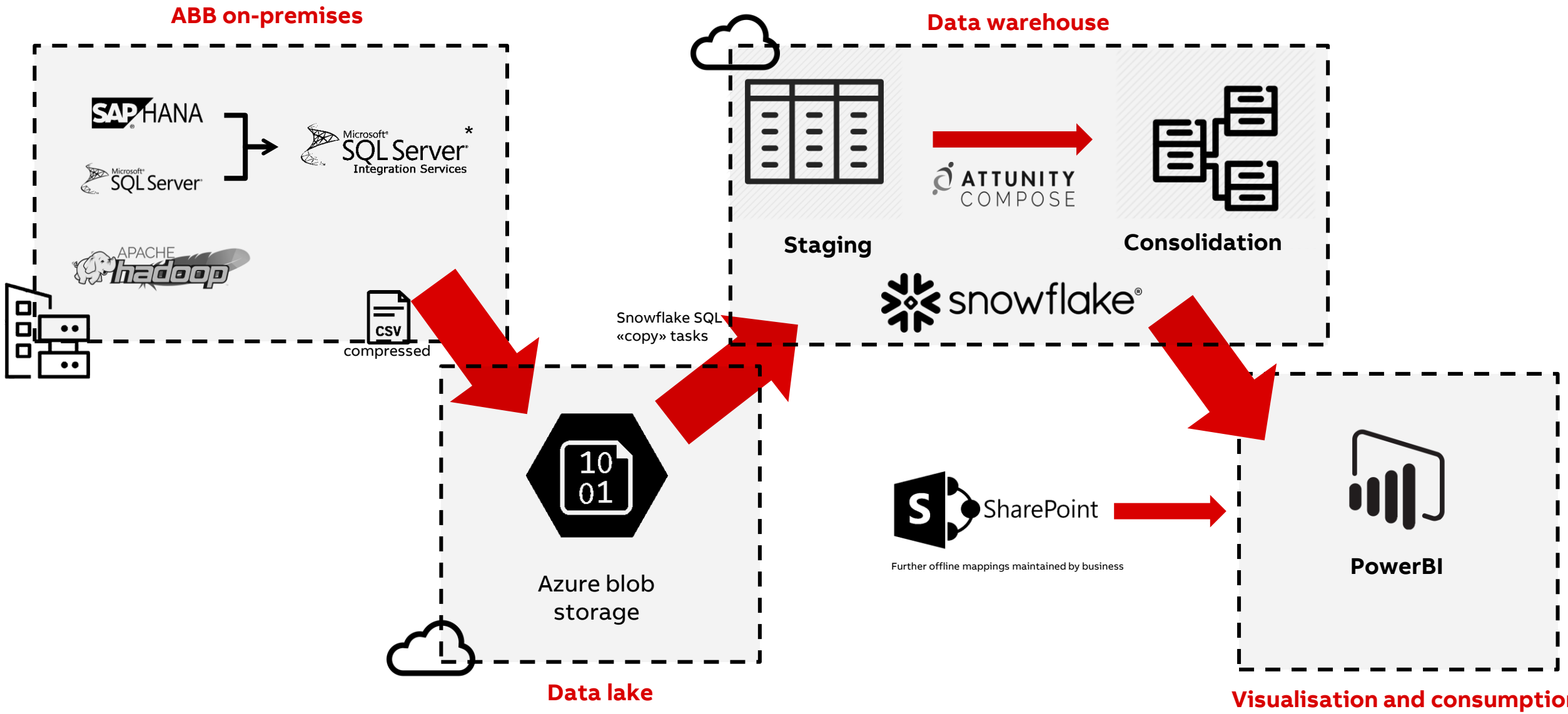
Il nuovo cruscotto powerbi è collegato a snowflake, che pesca da SAP HANA e Hadoop, rispettivamente collegati usando un job di SSIS che genera un csv zippato che viene poi spinto su azure blob. Da azure blob, c'è un ulteriore task snowflake che cancella la change table in staging e poi copia i dati dal blob su staging.

Dunque, un job di qlik compose inserisce/aggiorna i dati nella tabella di consolidation, da cui sono create due viste poi lette nel dashboard powerbi. Per Hadoop il processo è simile, ma il csv non viene generato tramite ssis bensì tramite un edge node di hadoop.

Questa nuova architettura è meglio spiegata nella prossima slide.



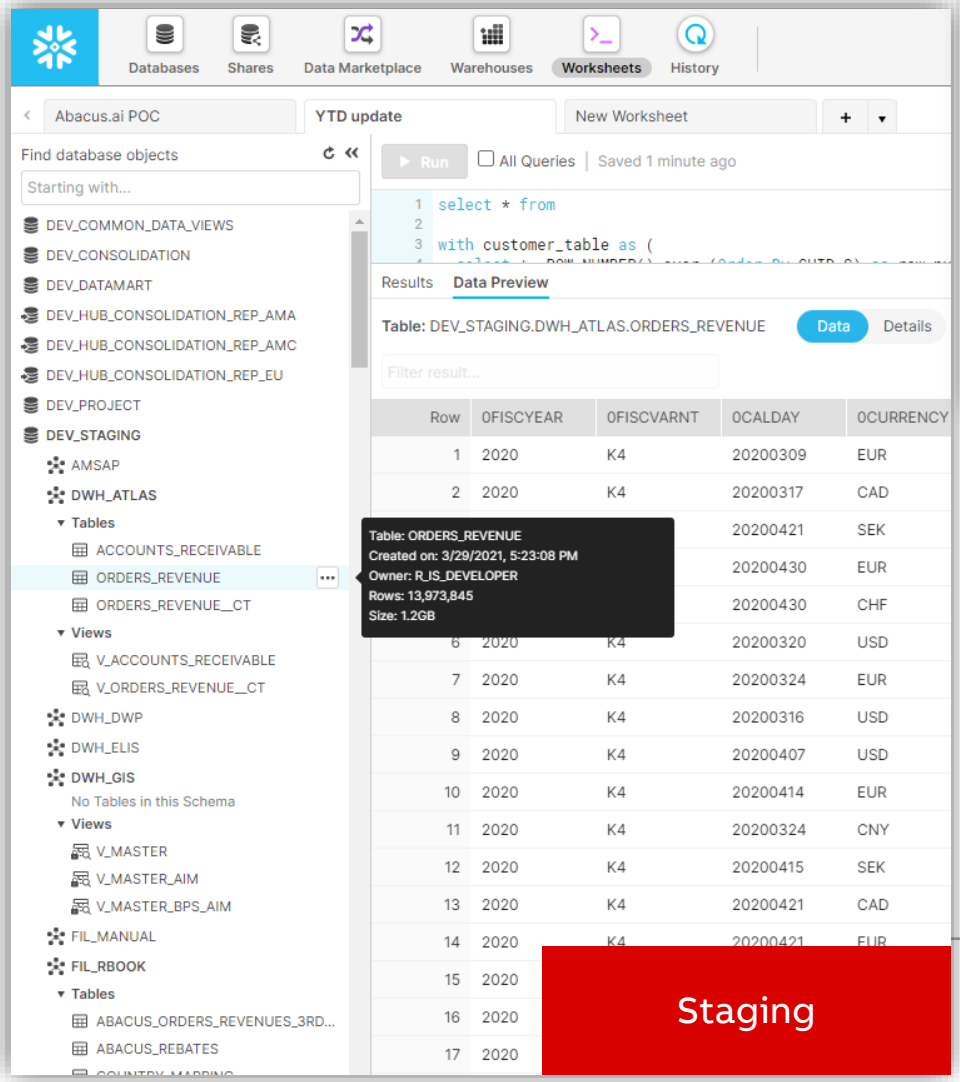
Architettura

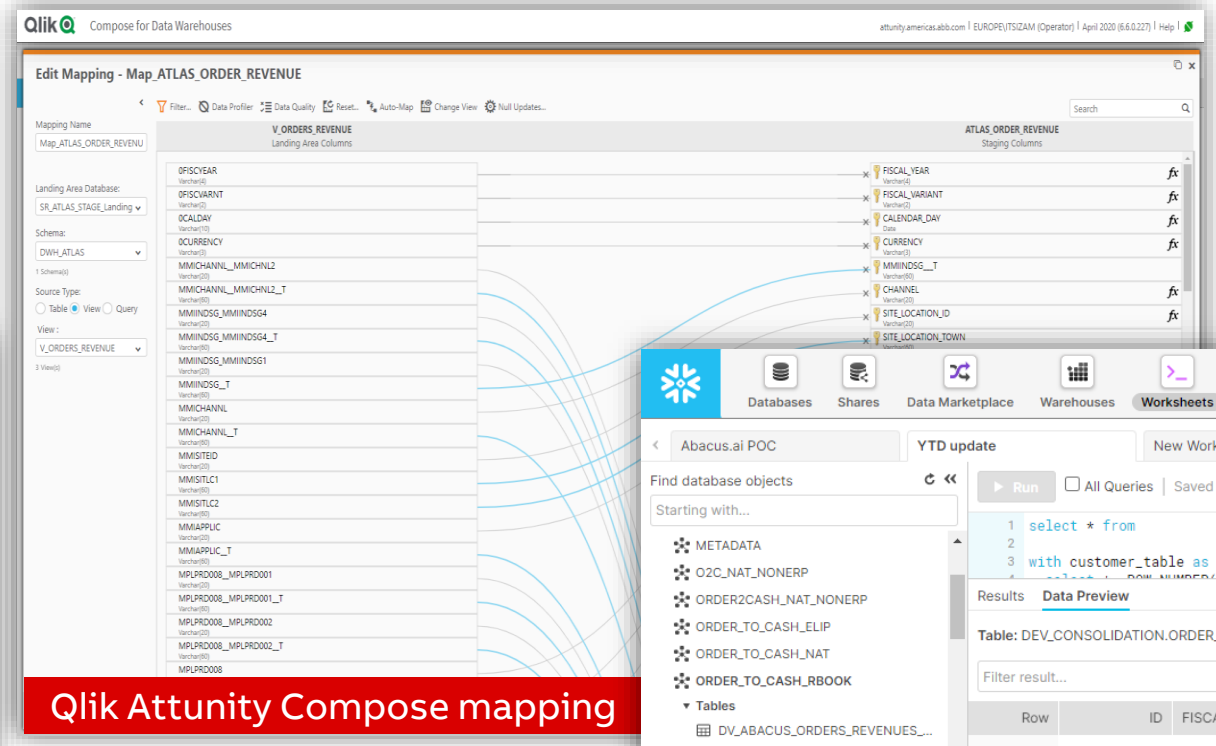


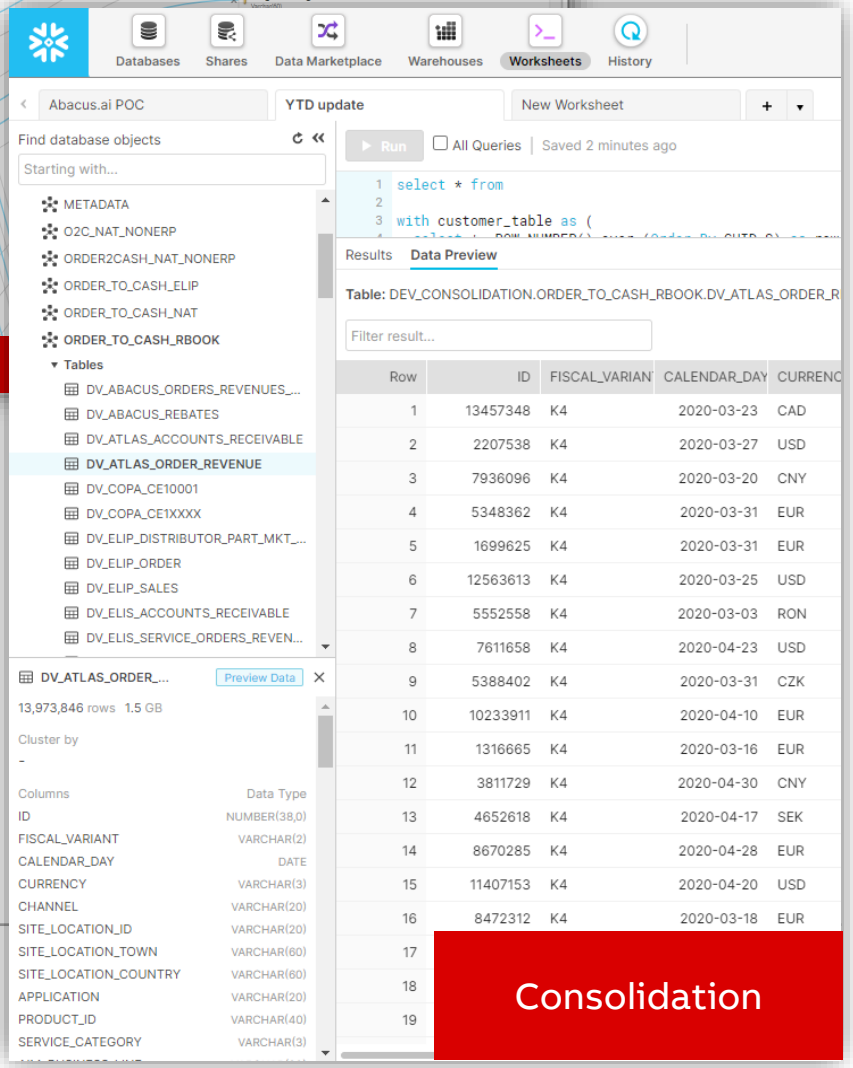
*pacchetto SSIS realizzato da sviluppatori di ABB

Qlik compose

Qlik compose + snowflake



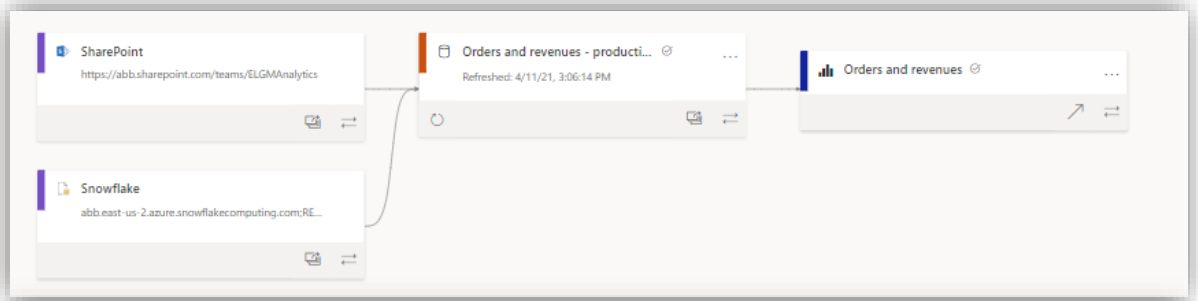


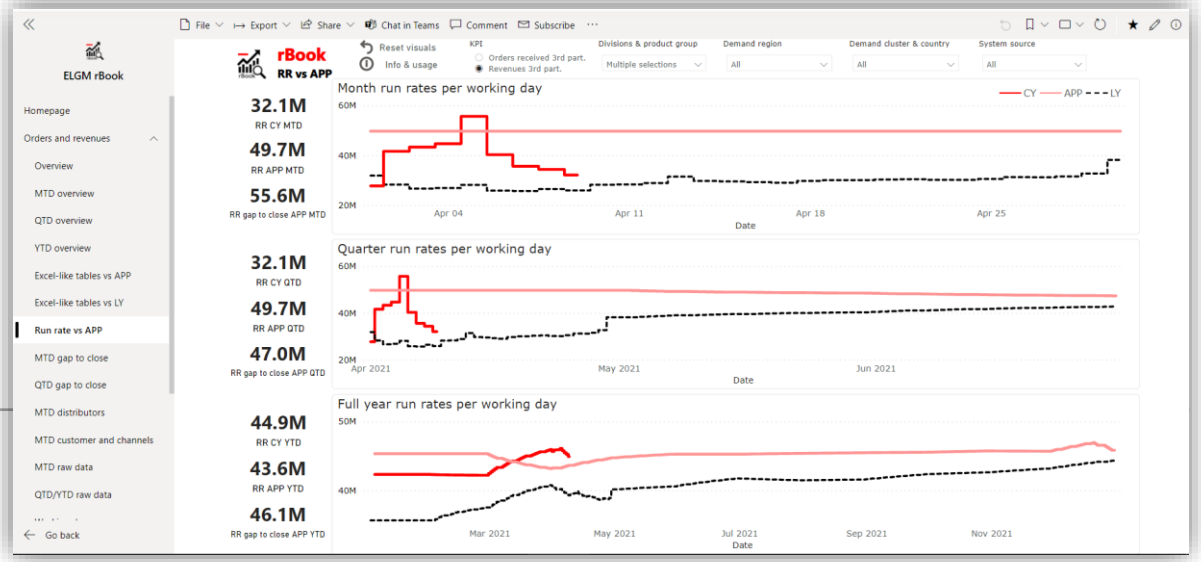


PowerBI

Visualizzazione e deployment in service

- Il layer di consolidation è stato dunque collegato a PowerBI insieme ad ulteriori mappature in Sharepoint. Il risultato su Powerbi service è un dataflow molto più semplice (primo screenshot in alto a destra).
- Per la visualizzazione, mi sono affidato a visual nativi di powerbi, per una migliore interattività. Ordinato e fatturato sono stati visualizzati secondo le seguenti principali dimensioni:
  - Business unit
  - Cliente
  - Regione e paese di appartenenza
  - Unità legale di vendita
  - Budget
- Numerose «measures» sono state create per effettuare benchmarking vs LY, vs budget o per visualizzare i trend quotidiani (screenshot in basso a destra).
- In service, la dashboard si rinfresca tutti i giorni alle 14 con un failsafe alle 15

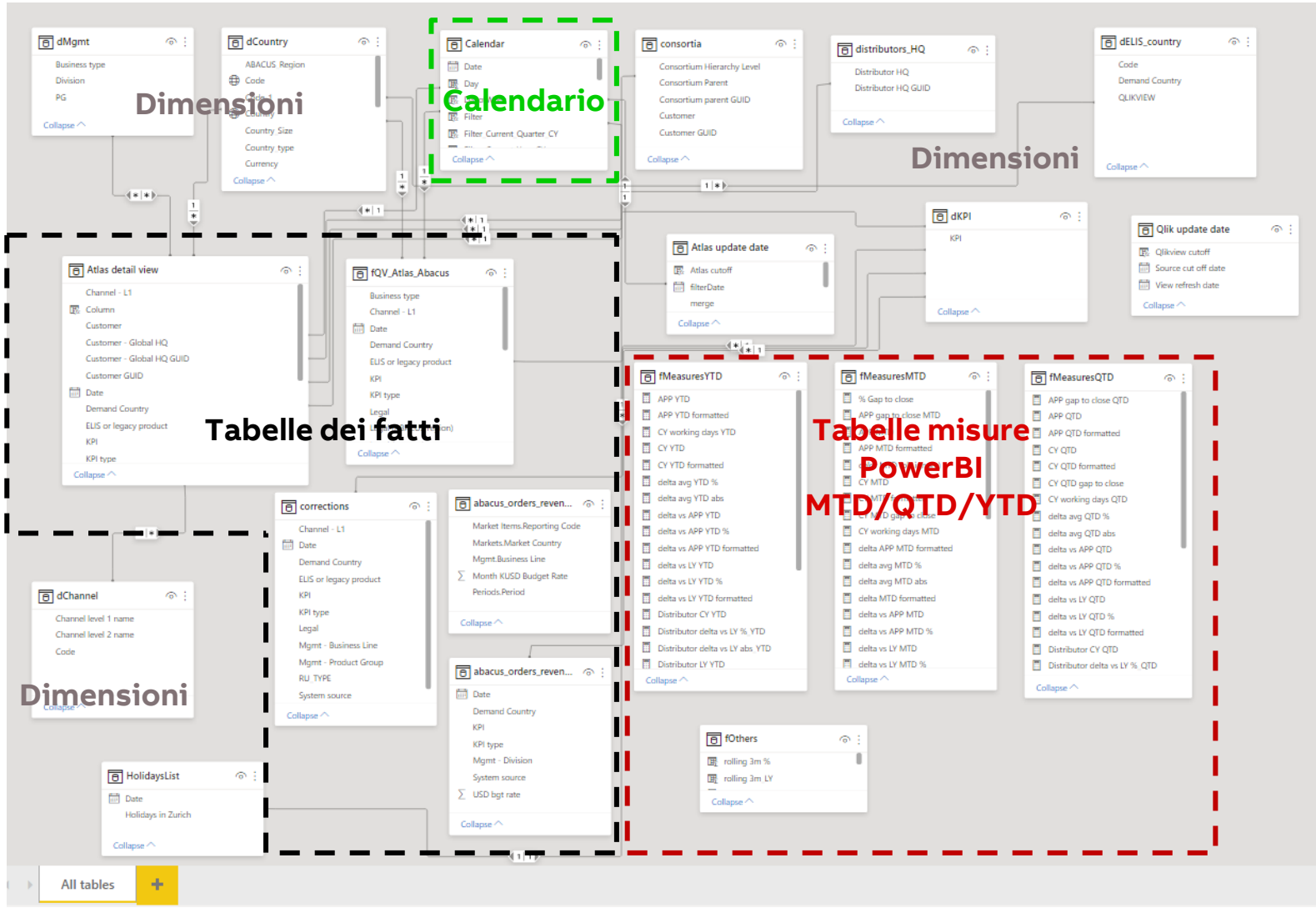




Data model

Dashboard PowerBI

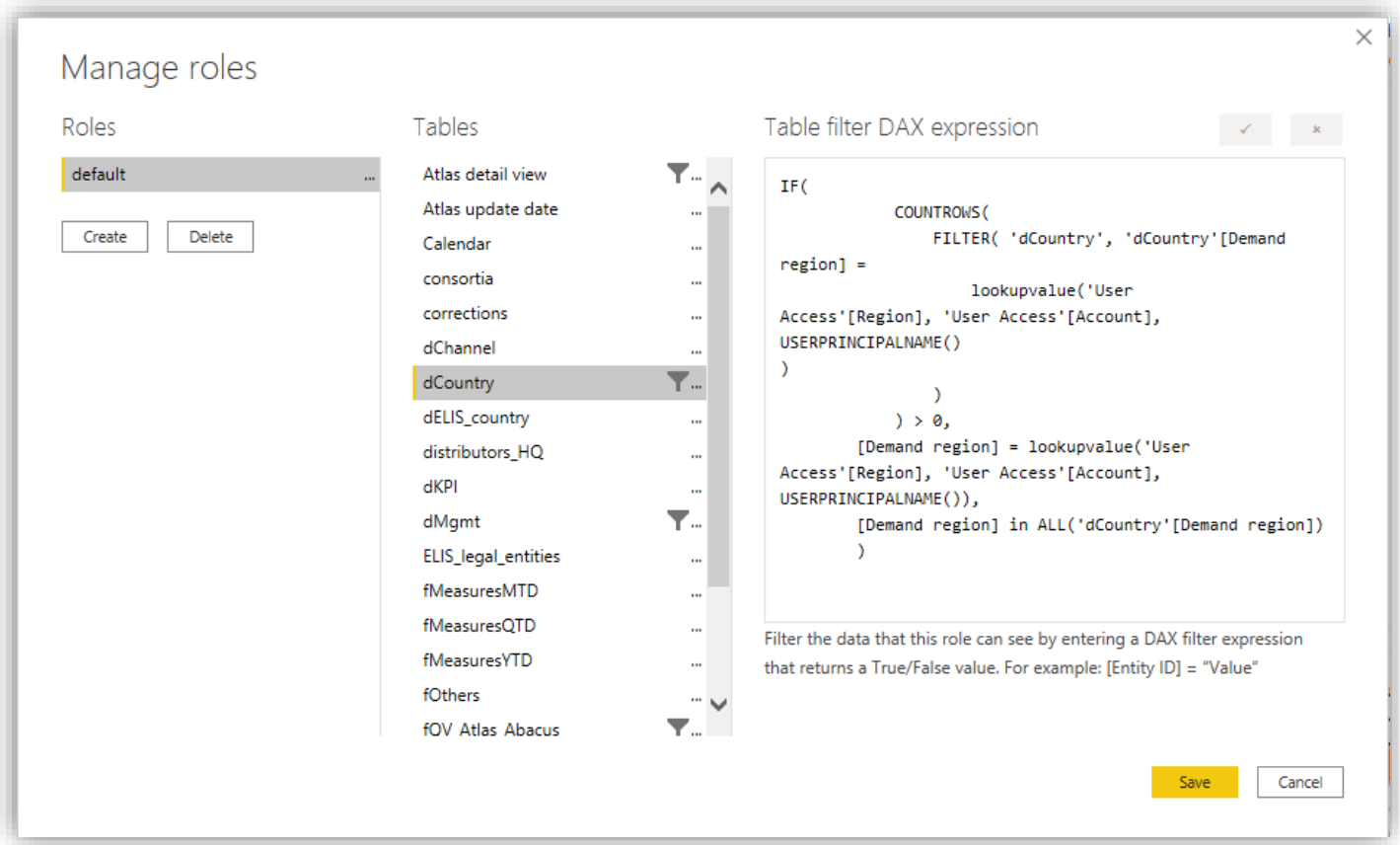
- A sinistra uno screenshot del servizio di analysis services di powerbi, ad esempio:
  - Dimensioni:
    - dKPI è la Dimensioni: ad esempio, dimensione per selezionare ordinato/fatturato
    - dMgmt: dimensioni per business unit
  - Misure:
    - Benchmarking vs LY e Budget
    - Filtri per canale di vendite
    - MTD/QTD/YTD
  - Tabelle dei fatti: transazioni per ordinato / fatturato, più eventuali correzioni



Row level access

Limitazione accesso utenti

- La dashboard è stata rilasciata a circa 400 utenti
- Per limitare l'accesso per i singoli utenti a livello di business unit / regione, è stato creato un ruolo di default che filtra sulle tabelle di dimensione
- In questo modo, gli utenti non hanno accesso a regioni o business unit diverse a quelle di appartenenza.
- Alcuni utenti hanno comunque accesso a livello globale.

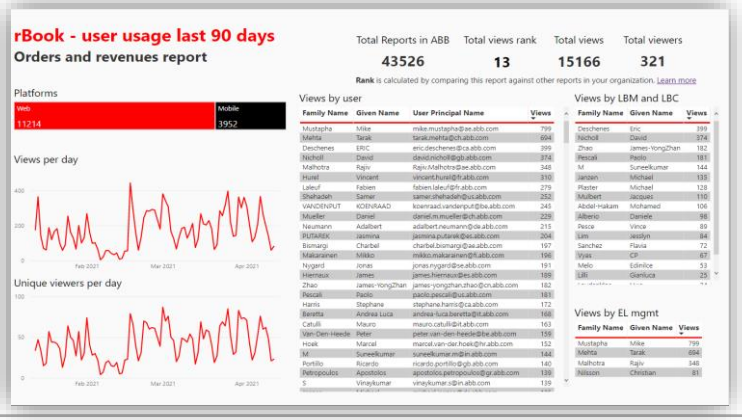


Conclusioni

Risultati, takeaways e next steps

Risultati e takeaways

- La dashboard è stata un successo, con 400+ utenti nella distribution list e di questi più di 300 sono attivi. Per questo una piccola dashboard per tracciare lo usage è stata creata in powerbi service* (come per screenshot sottostante). Al momento è la 13esima dashboard più utilizzata in ABB.
- Ci sono stati alcuni problemi con la stabilità del server di attunity e SSIS che poi sono stati stabilizzati con l'aiuto di sviluppatori senior



Next steps

- Non tutti i dati ingested in snowflake sono stati utilizzati per la realizzazione della dashboard.
  - ➢ Action: aumentare la mole di dati ingested da PowerBI utilizzando «incremental refresh» per visualizzare i dati fino a livello di singolo codice prodotto
- Migrare le «mappature» in sharepoint su blob e renderle disponibile in snowflake per altri progetti
- Il target è CPM per capire «cosa è successo e sta accedendo», in futuro si può prevedere l'integrazione nella stessa dashboard del forecasting POC, che vedremo nella prossima sezione

* Il report è semplicemente una modifica estetica del report «usage metrics» builin in powerbi service, purtroppo non scaricabile.

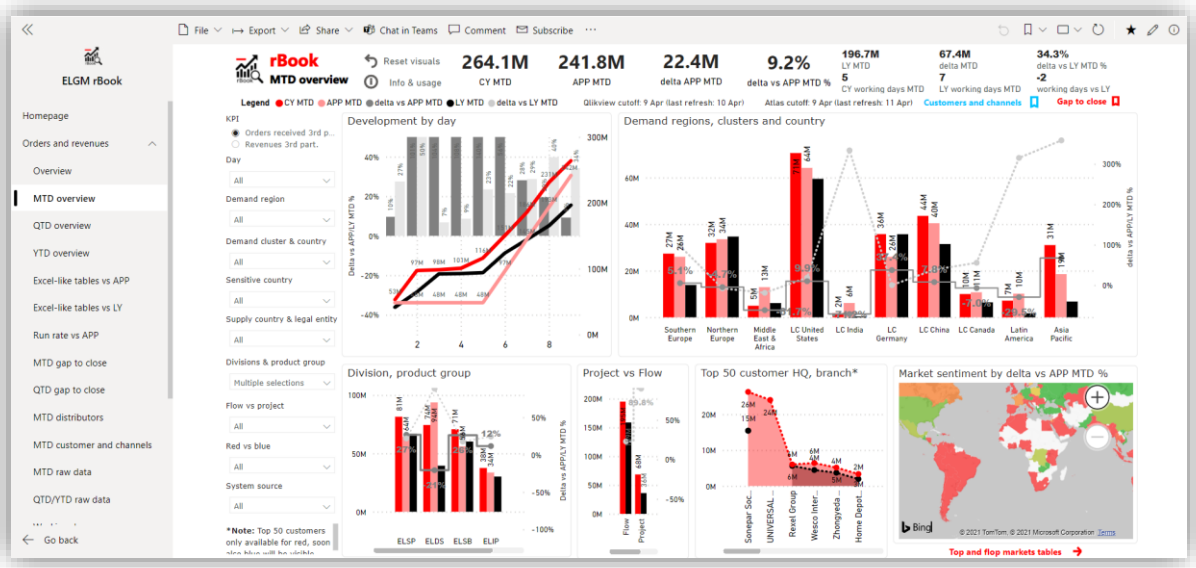
Advanced analytics

POC forecast

ABB Electrification business

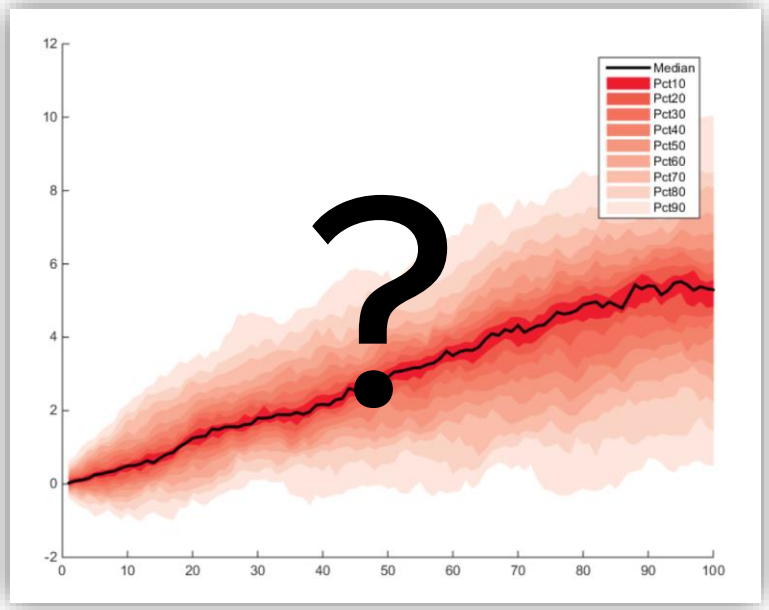
Decision making support per marketing and sales, finance

Reporting dashboard – presente e passato



La dashboard realizzata nella sezione precedente consente CPM per cosa è successo e cosa succederà

Predictive analytics – futuro



Al momento, per marketing e sales del business electrification, nessuno strumento di predictive analytics è in produzione

Forecasting POC

Architettura e dataset

Sources

Consolidation

snowflake

+

Azure blob storage

ML modelling + libraries

databricks

PYTORCH PROPHET

Visualisation

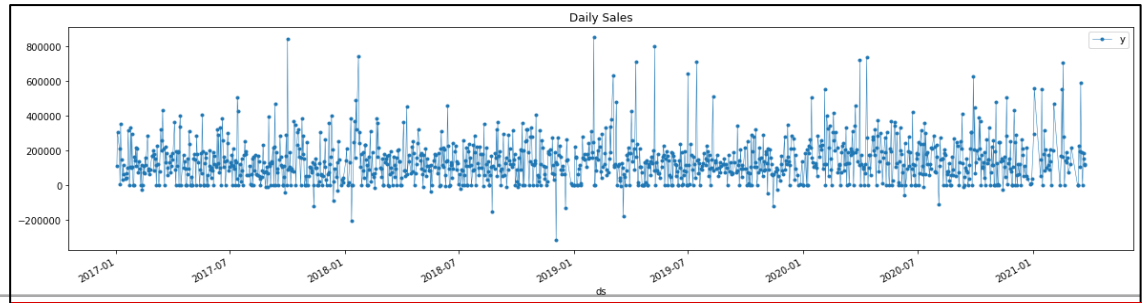
matplotlib

Architettura

Input – Serie storica dalle tabelle create in snowflake consolidation o se non presente, direttamente da blob storage. Il POC si limita alla serie storica di una product line

Modelling – databricks utilizzando pytorch e prophet

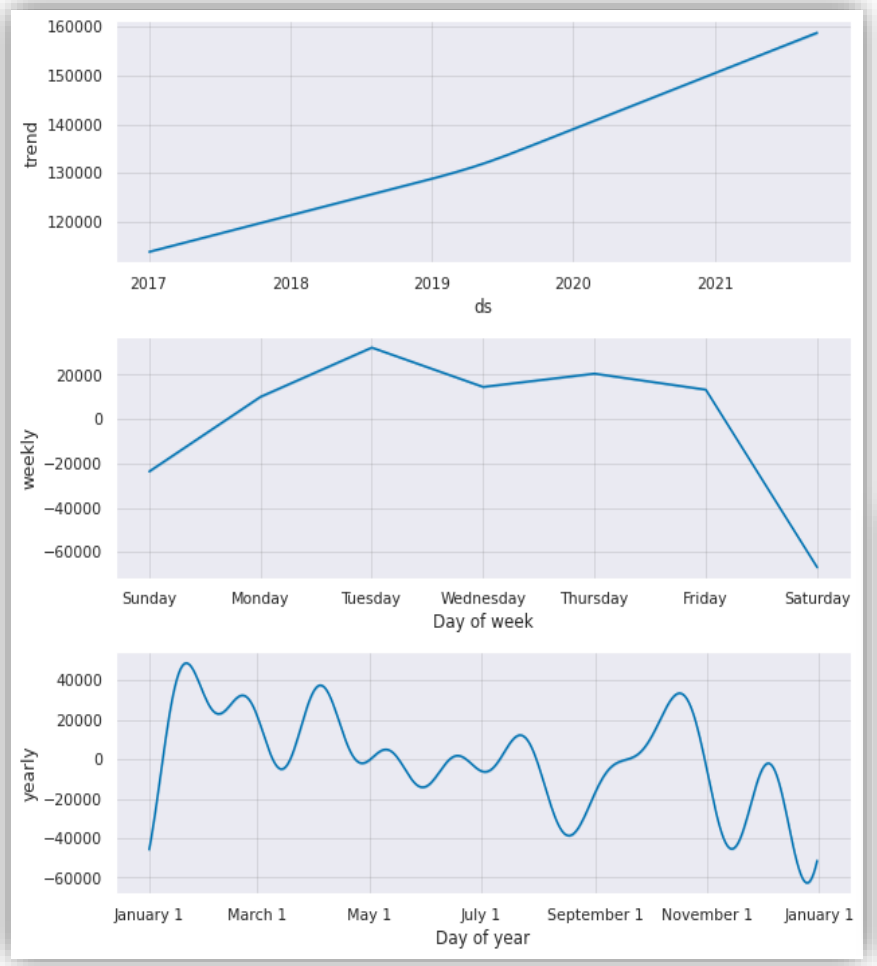
Output - Visualizzazione con matplotlib



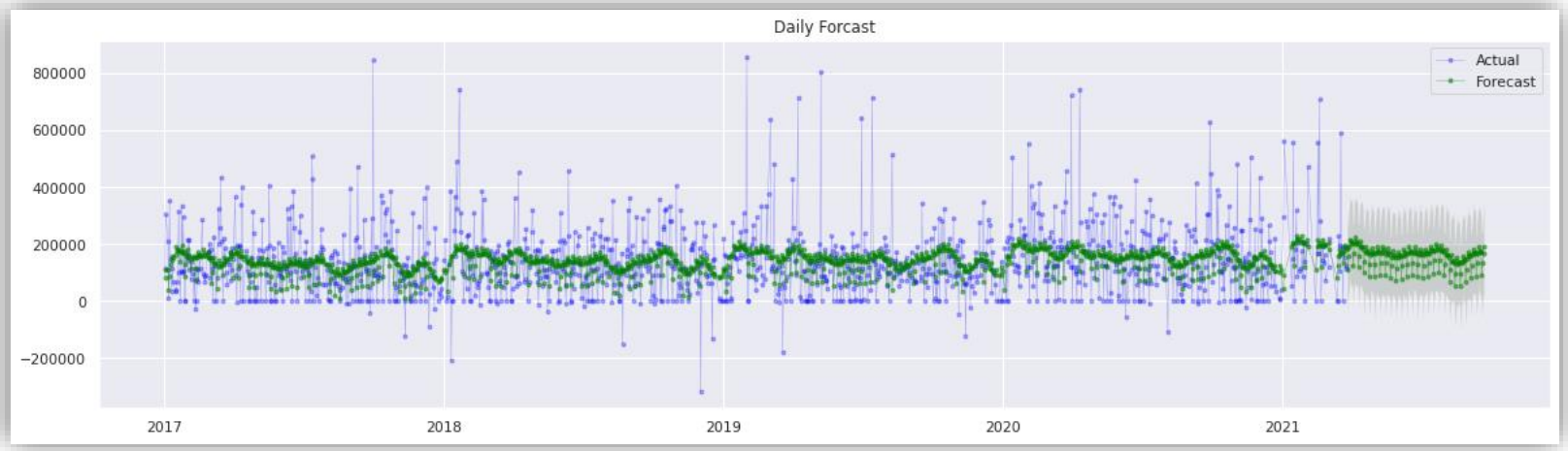
Il dataset di input è l'ordinato giornaliero degli ultimi 4 anni

Facebook prophet

Timeseries analysis



- Applicando Fbprophet sul dataset, si ottengono gli effetti di seasonality riportati a sinistra.
- Dunque, plottando la prediction per i prossimi sei mesi a livello giornaliero, come da grafico sottostante, si può osservare la prediction e lower/upper boundaries.
- Processando il dataset, sono stati rimossi gli outliers ma per la visualizzazione sono stati comunque inclusi.



Pytorch

LSTM for time series

- Per il POC, un'ulteriore obbiettivo è stato applicare un modello di LSTM per il forecast della stessa time series
- L'input al modello è stato semplificato, dal momento che è stata usata una matrice mensile e lag 12 mesi anziché giornaliera, altrimenti sarebbe stata troppo complessa per il tempo a disposizione.
- L'errore sul test è di 80KUSD, che comparato ad una media di 3.2MUSD è accettabile.
- Tramite matplotlib, è stato plottato forecast* per i prossimi 6 mesi e actual

Microsoft Azure | Databricks

blob_connection (Python)

spark 3 | File | Edit | View: Standard | Permissions | Run All | Clear

Command took 0.02 seconds -- by simone.zambetti@it.abb.com at 4/12/2021, 12:23:50 PM on spark 3

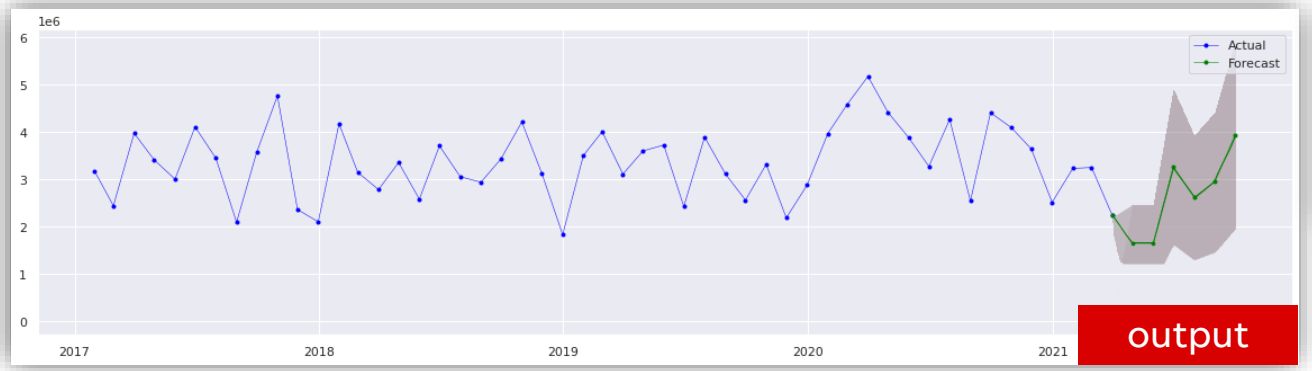
Cmd 27

```
num_timesteps, df_train = create_ts_df(df_model_all['y'].values, start_index=0, end_index=None, history_length=12, step_size=1, target_step=1)
df_train
```

Out[23]:

	x_lag13	x_lag12	x_lag11	x_lag10	x_lag9	x_lag8	x_lag7	x_lag6	x_lag5	x_lag4	x_lag3	x_lag2	y
0	3168841	2437942	3972875	3408975	3002092	4107383	3446743	2089842	3567098	4764357	2358110	2099227	4173700
1	2437942	3972875	3408975	3002092	4107383	3446743	2089842	3567098	4764357	2358110	2099227	4173700	3144439
2	3972875	3408975	3002092	4107383	3446743	2089842	3567098	4764357	2358110	2099227	4173700	3144439	2781623
3	3408975	3002092	4107383	3446743	2089842	3567098	4764357	2358110	2099227	4173700	3144439	2781623	3353178
4	3002092	4107383	3446743	2089842	3567098	4764357	2358110	2099227	4173700	3144439	2781623	3353178	2578602
5	4107383	3446743	2089842	3567098	4764357	2358110	2099227	4173700	3144439	2781623	3353178	2578602	3706293
6	3446743	2089842	3567098	4764357	2358110	2099227	4173700	3144439	2781623	3353178	2578602	3706293	3053330
7	2089842	3567098	4764357	2358110	2099227	4173700	3144439	2781623	3353178	2578602	3706293	3053330	2937281
8	3567098	4764357	2358110	2099227	4173700	3144439	2781623	3353178	2578602	3706293	3053330	2937281	3433900
9	4764357	2358110	2099227	4173700	3144439	2781623	3353178	2578602	3706293	3053330	2937281	3433900	4211275
10	2358110	2099227	4173700	3144439	2781623	3353178	2578602	3706293	3053330	2937281	3433900	4211275	3118584
11	2099227	4173700	3144439	2781623	3353178	2578602	3706293	3053330	2937281	3433900	4211275	3118584	1827712
12	4173700	3144439	2781623	3353178	2578602	3706293	3053330	2937281	3433900	4211275	3118584	1827712	3503383
13	3144439	2781623	3353178	2578602	3706293	3053330	2937281	3433900	4211275	3118584	1827712	3503383	4005571
14	2781623	3353178	2578602	3706293	3053330	2937281	3433900	4211275	3118584	1827712	3503383	4005571	3109469
15	3353178	2578602	3706293	3053330	2937281	3433900	4211275	3118584	1827712	3503383	4005571	3109469	3599585
16	2578602	3706293	3053330	2937281	3433900	4211275	3118584	1827712	3503383	4005571	3109469	3599585	3722557
17	3706293	3053330	2937281	3433900	4211275	3118584	1827712	3503383	4005571	3109469	3599585	3722557	2423188
18	3053330	2937281	3433900	4211275	3118584	1827712	3503383	4005571	3109469	3599585	3722557	2423188	3891825
19	2937281	3433900	4211275	3118584	1827712	3503383	4005571	3109469	3599585	3722557	2423188	3891825	3113990

Input monthly matrix



output

```
from sklearn.metrics import mean_absolute_error
print('test mean absolute error: {}'.format(mean_absolute_error(y_act, y_pred)))
```

```
test mean absolute error: 78236.25
```

```
Command took 0.02 seconds -- by simone.zambetti@it.abb.com at 4/12/2021, 12:23:51 PM on spark 3
```

Error on test

*l'intervallo di confidence interval è stato plottato deterministicamente moltiplicando 0.1, 0.2 e 0.5 per il point estimate. Avendo più tempo sarebbe stato meglio utilizzare dropout method

Repo

https://github.com/szambetti/BIBDA-final-project-public

Slide 17