



UNIVERSITA' DEGLI STUDI DI  
NAPOLI FEDERICO II

Scuola Politecnica e delle Scienze di Base  
Corso di Laurea Magistrale in Ingegneria Informatica

Information Systems and Business Intelligence

## ***Dashboard - "temp\_humid\_data"***

Anno Accademico [2023-2024]

*Prof.ssa Flora Amato*

Studenti  
**Riccardo Romano**  
matr. M63001489

**Giovanni Riccardi**  
matr. M63001480

**Francesco Panariello**  
matr. M63001433

# Indice

---

|  |    |
|--|----|
| Indice.....                              | II |
| Traccia Elaborato Dashboard.....         | 3  |
| Scopo Analisi .....                      | 4  |
| Metodologia di Sviluppo.....             | 4  |
| Struttura Dashboard .....                | 6  |
| Osservazione giornaliera.....            | 6  |
| Filtri visualizzazione .....             | 7  |
| Visualizzazione dati .....               | 8  |
| Trend del numero di Adult Males .....    | 9  |
| Statistiche Descrittive .....            | 11 |
| Mappa di Calore delle Correlazioni ..... | 12 |
| Predizione sui dati.....                 | 13 |
| Conclusioni .....                        | 14 |

## Traccia Elaborato Dashboard

---

Dashboard su Dataset Allegato:

- Sviluppare una dashboard interattiva
- La dashboard dovrebbe visualizzare informazioni chiave, tendenze e insight estratti dai dati.
- Considerare l'uso di grafici, tabelle, e mappe interattive.
- Presentare i dati in modo che siano intuitivi e informativi per gli utenti finali.

## Scopo Analisi

---

Questo studio presenta un'analisi condotta su un dataset denominato "temp\_humid\_data". Esso comprende dati relativi alla temperatura, all'umidità e alla presenza di adult males, raccolti durante un arco temporale di due anni.

L'obiettivo principale di questo studio è trasformare le informazioni raccolte in un'esperienza visiva e interattiva attraverso una dashboard. La dashboard fornirà agli utenti la possibilità di esplorare le relazioni tra temperatura, umidità e presenza maschile adulta nel corso del tempo, evidenziando tendenze e offrendo insights su possibili correlazioni.

Attraverso l'utilizzo di visualizzazioni dinamiche e strumenti interattivi, intendiamo rendere accessibili le informazioni complesse contenute nel dataset. Gli utenti saranno dunque in grado di analizzare i dati in modo più approfondito, identificare pattern e formulare ipotesi sulla base delle relazioni emerse.

### *Metodologia di Sviluppo*

La dashboard è stata sviluppata in python, con l' utilizzo della libreria streamlit.

Streamlit è una libreria Python progettata per semplificare la creazione rapida di dashboard interattive per la visualizzazione dei dati. Grazie alla sua sintassi minimalista, consente agli sviluppatori di trasformare script Python in applicazioni web con widget interattivi, facilitando la creazione di visualizzazioni dati coinvolgenti senza richiedere conoscenze avanzate di sviluppo web.

Il dataset "temp\_humid\_data" è suddiviso in due componenti principali: "Sheet1" e "Sheet3", i quali si differenziano soprattutto per il fatto di essere selezionati su due annate differenti, ovvero

2022 per il primo e 2023 per il secondo, per il 2023 è presente anche il numero di Adult Males, non presente nei dati dell' anno precedente.

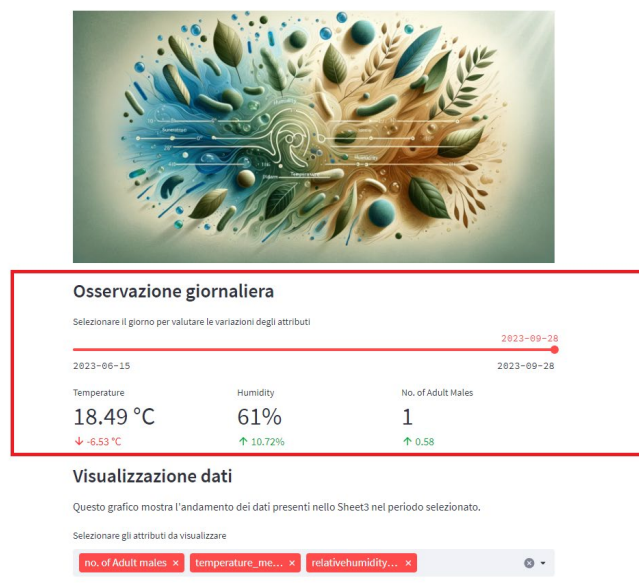
La dashboard presenta in maniera interattiva i dati relativi al 2023 (Sheet3), mostrando dunque le relazioni fra temperatura ed umidità e la presenza di Adult Males nella popolazione batterica.

# Struttura Dashboard

In questo capitolo vengono presentate le diverse sezioni della Dashboard e le loro funzionalità.

## Osservazione giornaliera

Nella pagina principale della Dashboard è presente la sezione per osservare i Dati in uno specifico giorno.



Per la visualizzazione dei dati la schermata è stata divisa in 3 colonne ognuna contenente un elemento *metrics* di streamlit, in cui è mostrato il valore del dato e la variazione rispetto la media.

```
col1, col2, col3 = st.columns(3)

if not fil_data.empty:
    temperature_mean = data['temperature_mean'].mean()
    humidity_mean = data['relativehumidity_mean'].mean()
    adult_males_mean = data['no. of Adult males'].mean()

    # Calculate the variations
    temperature_variation = fil_data['temperature_mean'].values[0] - temperature_mean
    humidity_variation = fil_data['relativehumidity_mean'].values[0] - humidity_mean
    adult_males_variation = fil_data['no. of Adult males'].values[0] - adult_males_mean

    col1.metric("Temperature", f"{fil_data['temperature_mean'].values[0]} °C",
               f"{temperature_variation:.2f} °C")
    col2.metric("Humidity", f"{fil_data['relativehumidity_mean'].values[0]}%",
               f"{humidity_variation:.2f}%")
    col3.metric("No. of Adult Males", f"{fil_data['no. of Adult males'].values[0]}",
               f"{adult_males_variation:.2f}")
```

Per la selezione del giorno da osservare è stato utilizzato l'elemento slider che restituisce l'oggetto `fil_data` contenente i dati del giorno selezionato.

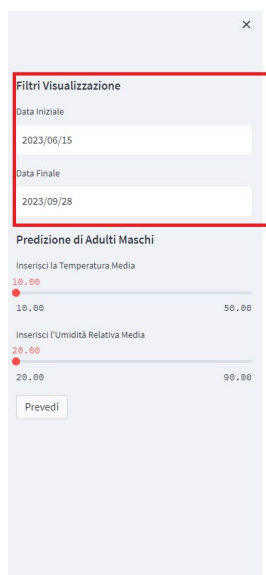
```
st.subheader("Osservazione giornaliera")
data2 = load_data()
data2['Date'] = pd.to_datetime(data['Date'])
date_time = st.slider(
    "Selezionare il giorno per valutare le variazioni degli attributi",
    value=(datetime(2023, 6, 15)),
    format_="__" "YYYY-MM-DD",
    min_value=(data2['Date'].min()),
    max_value=(data2['Date'].max())
)

# Filter data for the selected date
fil_data = data[data2['Date'] == date_time]
```

## Filtri visualizzazione

Nella sidebar è presente la sezione per filtrare i dati, permettendo la visualizzazione, nella dashboard, del solo periodo selezionato.

Lo scopo del filtraggio può essere un'analisi approfondita di un determinato periodo dell'anno, o di un determinato mese.



Esplora i dati e predici il numero di adulti maschi in base a temperatura e umidità per Sheet3 - temp\_humid\_data



### Osservazione giornaliera

Selezionare il giorno per valutare le variazioni degli attributi

2023-06-15 2023-09-28

| Temperature | Humidity | No. of Adult Males |
|-------------|----------|--------------------|
| 18.49 °C    | 61%      | 1                  |
| ↓ -6.53 °C  | ↑ 10.72% | ↑ 0.58             |

Visualizzazione dati

Consiste in due elementi `date_input`, che permettono la selezione della data iniziale e della data finale, restituisce poi l'elemento `filtered_data`, che sarà utilizzato nelle altre sezioni della Dashboard

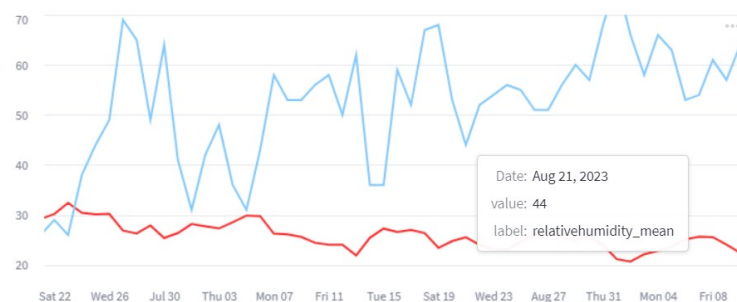
```
# Sidebar per filtri
st.sidebar.subheader("Filtri Visualizzazione")
start_date = st.sidebar.date_input("Data Iniziale", data['Date'].min().date())
end_date = st.sidebar.date_input("Data Finale", data['Date'].max().date())
# Filtra i dati in base all'intervallo di date selezionato
start_date = pd.to_datetime(start_date)
end_date = pd.to_datetime(end_date)
filtered_data = data[(data['Date'] >= start_date) & (data['Date'] <= end_date)]
```

## Visualizzazione dati

In questa sezione è presente un grafico che mostra l'andamento dei dati presenti nel dataset nel periodo scelto.



Per la visualizzazione del grafico è stato utilizzato `line_chart` di streamlit, che consente la generazione di un grafico a linee interattivo, è infatti possibile zoomare e muoversi nel grafico per esplorare dettagliatamente i segmenti.





È inoltre possibile selezionare tramite un multiselect quali attributi visualizzare all' interno del grafico

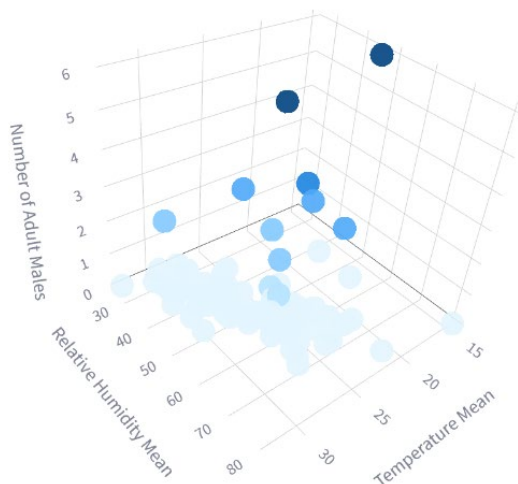


### *Trend del numero di Adult Males*

Questa sezione offre uno studio specifico sul numero di adult males presenti nel periodo selezionato.

Il primo grafico utile per lo studio del numero di adult males è il grafico di dispersione 3D, il quale ci permette una visualizzazione chiara e tridimensionale dei dati per trovare una possibile relazione tra temperatura, umidità e numero di Adult Males.

Grafico di dispersione 3D degli attributi



Per la realizzazione del grafico è stata utilizzata la libreria Plotly, in particolare con la classe go.Figure è stato creato un oggetto di tipo figura, ed aggiunto un trace di tipo Scatter3d.

Le colonne "temperature\_mean", "relativehumidity\_mean", e "no. of Adult males" sono state utilizzate come coordinate x, y e z rispettivamente. Ogni punto è rappresentato da un marker con dimensioni, colore e opacità variabili in base al numero di Adult Males associato.

```
# Creazione di un grafico di dispersione 3D
scatter_3d = go.Figure()
scatter_3d.add_trace(go.Scatter3d(
    x=filtered_data['temperature_mean'],
    y=filtered_data['relativehumidity_mean'],
    z=filtered_data['no. of Adult males'],
    mode='markers',
    marker=dict(size=10, color=filtered_data['no. of Adult males'], opacity=0.9),
    text=filtered_data['no. of Adult males'],
))
scatter_3d.update_layout(scene=dict(
    xaxis_title='Temperature Mean',
    yaxis_title='Relative Humidity Mean',
    zaxis_title='Number of Adult Males',
),
    margin=dict(l=0, r=0, b=0, t=0),
)
st.plotly_chart(scatter_3d)
```

Il grafico a barre per la visualizzazione del numero di Adult Males permette di sapere quali sono i giorni in cui ci sono stati Adult Males ed il numero di Adult Males presenti in quei giorni, supportato dalla tabella riassuntiva, per visualizzarne anche gli altri attributi.

Visualizzazione del numero di adulti maschi nel periodo selezionato.



Focus sui giorni con presenza di Adult Males

|    | Date                | no. of Adult males | temperature_mean | relativehumidity_mean |
|----|---------------------|--------------------|------------------|-----------------------|
| 28 | 2023-07-13 00:00:00 | 3                  | 25.2             | 41                    |
| 30 | 2023-07-15 00:00:00 | 2                  | 29.64            | 31                    |
| 33 | 2023-07-18 00:00:00 | 1                  | 25.63            | 54                    |
| 74 | 2023-08-28 00:00:00 | 2                  | 25.59            | 56                    |
| 76 | 2023-08-30 00:00:00 | 1                  | 25.67            | 57                    |
| 84 | 2023-09-07 00:00:00 | 6                  | 25.67            | 54                    |
| 86 | 2023-09-09 00:00:00 | 4                  | 24.08            | 57                    |
| 87 | 2023-09-10 00:00:00 | 3                  | 22.41            | 64                    |
| 89 | 2023-09-12 00:00:00 | 3                  | 22.04            | 53                    |
| 90 | 2023-09-13 00:00:00 | 2                  | 23.88            | 47                    |

I dati sono dunque stati filtrati, per ottenere solo i punti d' interesse (No. Adult Males > 0), stampati sottoforma di tabella ed è stato creato il grafico con l' elemento di streamlit *bar\_chart*.

```
# Creazione dei dati per il grafico a barre
plot_data = filtered_data.iloc[:, 1:] # Mostra solo ogni quinta data
bar_chart_data = plot_data[['Date', 'no. of Adult males']].set_index('Date')
filtered_plot_data = plot_data[plot_data['no. of Adult males'] > 0]
st.write("Focus sui giorni con presenza di Adult Males")
# Visualizza solo i dati filtrati
st.write(filtered_plot_data)
# Creazione del grafico a barre con st.bar_chart di Streamlit
st.write("Visualizzazione del numero di adulti maschi nel periodo selezionato.")
st.bar_chart(bar_chart_data, use_container_width=True)
```

## Statistiche Descrittive

Con la funzione *describe()* della libreria Pandas è stata inserita all'interno della dashboard una tabella riepilogativa delle statistiche descrittive del dataset nel periodo scelto.

Queste informazioni quali media, deviazione standard ed altri indicatori statistici sono utili in fase di analisi dei dati e nella visualizzazione di questi perché offrono una panoramica sintetica dei dati che si stanno analizzando ed aiutano ad identificare tendenze, outlier e modelli nei dati.

■ no. of Adult males

| Statistiche Descrittive                              |                    |                  |                       |
|--|--------------------|------------------|-----------------------|
| Statistiche descrittive delle variabili nel dataset. |                    |                  |                       |
|  | no. of Adult males | temperature_mean | relativehumidity_mean |
| count  | 90                 | 90               | 90                    |
| mean   | 0.3667             | 25.5831          | 49.8778               |
| std  | 1.1461             | 3.1441           | 11.7663               |
| min  | 0                  | 14.74            | 26                    |
| 25%  | 0                  | 24.0125          | 41                    |
| 50%  | 0                  | 25.685           | 51.5                  |
| 75%  | 0                  | 27.39            | 57.75                 |
| max  | 6                  | 32.41            | 81                    |

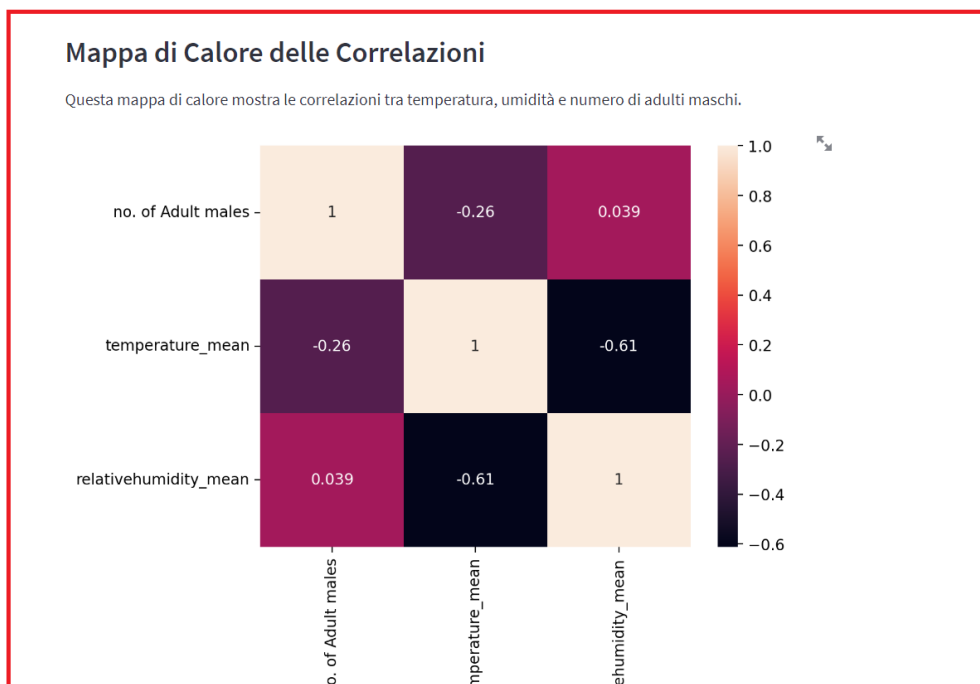
## Mapa di Calore delle Correlazioni

La mappa di calore è utile per visualizzare le correlazioni tra le variabili "temperatura", "umidità" e "numero di Adult Males" nei dati selezionati.

Una mappa di calore rappresenta graficamente le correlazioni tra coppie di variabili, evidenziando la forza e la direzione di tali correlazioni attraverso l'uso di colori.

Un coefficiente di correlazione:

- Vicino a 1 indica una forte correlazione positiva: quando una variabile aumenta, l'altra tende a aumentare.
- Vicino a -1 indica una forte correlazione negativa: quando una variabile aumenta, l'altra tende a diminuire.
- Vicino a 0 indica una debole o nessuna correlazione: le variazioni in una variabile non sono predittive delle variazioni nell'altra.



Per realizzare la mappa di calore è stata utilizzata la libreria *Seaborn*, con la funzione `heatmap` al quale è stata passata la correlazione calcolata sui dati.

```
# Mappa di calore per le correlazioni
st.subheader("Mappa di Calore delle Correlazioni")
st.write("Questa mappa di calore mostra le correlazioni tra temperatura, umidità e numero di adulti maschi.")
corr = filtered_data.corr()
fig, ax = plt.subplots()
sns.heatmap(corr, annot=True, ax=ax)
st.pyplot(fig)
```

## Predizione sui dati

Nella sidebar è presente una sezione per la predizione del numero di Adult Males rispetto ai valori di temperatura ed umidità selezionati.



Per effettuare la predizione è stato addestrato un modello di Regressione Lineare sui dati presenti nel dataset.

La regressione lineare è un metodo statistico utilizzato per modellare la relazione lineare tra una variabile dipendente (y) e una o più variabili indipendenti (x). L'obiettivo è trovare la linea di regressione che minimizza la somma dei quadrati delle differenze tra i valori osservati e quelli predetti. Questo modello può essere utilizzato per fare previsioni o comprendere la relazione tra le variabili coinvolte.

```
# Funzione per addestrare il modello di regressione
def train_model(data):
    X = data[['temperature_mean', 'relativehumidity_mean']]
    y = data['no. of Adult males']
    model = LinearRegression()
    model.fit(X, y)
    return model
```

## Conclusioni

---

In conclusione, nella realizzazione della dashboard interattiva in Streamlit, si evidenziano chiaramente i benefici distintivi offerti da questa piattaforma per la realizzazione di pagine web per la visualizzazione e analisi dei dati oggetto dello studio. Streamlit, con la sua sintassi intuitiva e la facilità di integrazione di widget interattivi, ha agevolato una rapida esplorazione delle relazioni tra temperatura, umidità e la presenza di Adult Males nel corso del periodo considerato.

La dashboard interattiva ha permesso una valutazione visiva delle variazioni dei dati, consentendo agli utenti di identificare facilmente pattern e trend rilevanti. L'interattività della dashboard ha fornito agli utenti uno strumento dinamico per esplorare i dati e formulare ipotesi in modo efficiente. Tuttavia, è importante notare che, a causa delle limitazioni del dataset, l'applicazione di modelli predittivi potrebbe essere vincolata e richiedere dati più completi per ottenere risultati affidabili.