



**Politecnico
di Torino**

POLITECNICO DI TORINO

Dipartimento di Ingegneria Gestionale e della Produzione

**Corso di Laurea Triennale
in Ingegneria Gestionale
Classe L-8 Ingegneria dell'Informazione**

A.A. 2024/2025

Software a supporto del processo decisionale

Relatore

Prof. Averta Giuseppe Bruno

Candidato

Contreras Tupac Yeny Andrea
Matricola: s259614

Marzo 2025

INDICE

1. PROPOSTA DI PROGETTO	3
1.1 STUDENTE PROPONENTE	3
1.2 TITOLO DELLA PROPOSTA.....	3
1.3 DESCRIZIONE DEL PROBLEMA PROPOSTO	3
1.4 DESCRIZIONE DELLA RILEVANZA GESTIONALE DEL PROBLEMA.....	3
1.5 DESCRIZIONE DEL DATA-SET PER LA VALUTAZIONE	3
1.6 DESCRIZIONE PRELIMINARE DEGLI ALGORITMI COINVOLTI	3
1.7 DESCRIZIONE PRELIMINARE DELLE FUNZIONALITÀ PREVISTE PER L'APPLICAZIONE	4
2. DESCRIZIONE DETTAGLIATA DEL PROBLEMA AFFRONTATO	5
2.1 DECISION MAKING	5
2.1.1 ANALISI DEI DATI	5
2.1.2 ANALISI CVP	6
2.2 L'IMPORTANZA DELL'AUTENTICAZIONE NELLE PMI	8
3. DESCRIZIONE DEL DATA-SET UTILIZZATO	9
3.1. STRUTTURA DEI DATI	9
3.2. DIAGRAMMA ER.....	10
4. DESCRIZIONE AD ALTO LIVELLO DELLE STRUTTURE DATI E DEGLI ALGORITMI COINVOLTI.....	11
4.1. STRUTTURA DEL PROGETTO	11
4.2. ALGORITMI UTILIZZATI NEL PROGETTO.....	12
4.2.1 ALGORITMO RICORSIVO	12
4.2.2 SIMULAZIONE.....	16
5. ALCUNE VIDEATE DELL'APPLICAZIONE E IL LINK AL VIDEO DIMOSTRATIVO DEL SOFTWARE.....	19
5.1. VIDEATA ALL'APERTURA DELL'APPLICAZIONE	19
5.2. VIDEATA RISULTATI	20
5.2. LINK YOUTUBE	25
6. VALUTAZIONI SUI RISULTATI OTTENUTI	26
6.1. CONSIDERAZIONI FINALI.....	26
7. LICENZA	27

1. PROPOSTA DI PROGETTO

1.1 Studente proponente

S259614 Contreras Tupac Yeny Andrea

1.2 Titolo della proposta

Software a supporto del processo decisionale

1.3 Descrizione del problema proposto

L'applicazione si propone di aiutare i responsabili dei negozi di giocattoli nella loro attività decisionale e di controllo, fornendo tramite grafici e tabelle, una panoramica generale delle performance per identificare informazioni indispensabili, ad esempio quale prodotto contribuisce maggiormente ai profitti, i trend, la redditività delle varie categorie di prodotti...

Verrà svolto anche un'analisi CVP (costo-volume-profitto) in modo che i manager possano conoscere la correlazione fra costo, volume e reddito e attraverso la simulazione potranno capire se attuare o meno determinate azioni correttive per raggiungere un certo risultato, ad esempio se diminuendo il prezzo di vendita e aumentando il budget pubblicitario è possibile aumentare le vendite del prodotto x.

1.4 Descrizione della rilevanza gestionale del problema

L'obiettivo è quello di sviluppare un'applicazione di supporto per i manager affinché siano guidati verso decisioni strategiche che favoriscano la crescita aziendale e, dove occorre, operazioni correttive per una gestione delle risorse più efficiente.

L'applicazione non si limiterà a fornire un quadro generale delle performance dei vari punti vendita ma verrà anche svolta una simulazione che aiuterà i manager a capire che impatto hanno certe variazioni sul reddito operativo.

1.5 Descrizione dei data-set per la valutazione

Il data-set utilizzato è stato reperito dal sito Kaggle.com (comunità di dati scientist in cui è possibile trovare e pubblicare set di dati), con una licenza di utilizzo non commerciale.

Esso è composto da quattro tabelle interconnesse che includono dati relativi ad una catena di negozi di giocattoli con diversi sedi in Messico chiamata Maven Toys.

Tale data-set include informazioni su prodotti, negozi, transazioni giornaliere e livelli attuali di inventario in ciascuna sede.

<https://www.kaggle.com/datasets/mysarahmadbhat/toy-sales>

1.6 Descrizione preliminare degli algoritmi coinvolti

Il Programma sarà sviluppato in linguaggio Python, utilizzando i pattern MVC (Model, View, Controller) e DAO per garantire la separazione tra l'interfaccia utente, la logica applicativa e l'accesso ai dati.

L'applicazione metterà a disposizione un menu a tendina affinché l'utente possa selezionare la location di interesse e tramite query SQL verranno ricercate le informazioni necessarie contenute nel data-set che saranno stampate poi in formato tabellare e grafico.

Attraverso un algoritmo ricorsivo verrà identificato il negozio e il prodotto più redditizio.

All'utente sarà anche richiesto di inserire in input dei dati di tipo numerico per poter implementare una serie di formule per l'analisi CVP e avviare la simulazione.

1.7 Descrizione preliminare delle funzionalità previste per l'applicazione software

L'obiettivo è quella di sviluppare un'applicazione che permette all'utente di:

- selezionare la location che si vuole analizzare in modo da poter visualizzare l'andamento delle vendite e consentire una migliore comprensione dei dati tramite un output anche di tipo grafico e tabellare
- inserire in input alcuni parametri necessari affinché il programma possa generare il diagramma di redditività che evidenzierà la dipendenza esistente tra ricavi, costi e volume di attività prodotto e avviare la simulazione per poter aver un'idea su eventuali scenari ipotetici

2. DESCRIZIONE DETTAGLIATA DEL PROBLEMA

2.1. Decision Making

Il Decision Making rappresenta un processo che conduce alla determinazione di una scelta da parte di un individuo o di un gruppo di persone.

In questa fase si identificano i problemi o le opportunità e si prendono decisioni basate su analisi e valutazioni.

La capacità di effettuare scelte ponderate e strategiche risulta cruciale per la crescita sostenibile e la gestione efficace di un'impresa.

Fattori influenzanti

Il processo decisionale, in ambito aziendale, è influenzato da una varietà di fattori interni e esterni tra cui la cultura organizzativa, le dinamiche di mercato e le normative vigenti, come la Legge Antitrust (Legge n.287/1990), che mira a prevenire pratiche commerciali sleali e a promuovere la concorrenza leale nel mercato, imponendo alle imprese di operare in conformità con i principi etici e legali.

Strumenti principali

Tra i principali strumenti utilizzati per migliorare l'efficacia decisionale e ridurre l'incertezza, e che l'applicazione in questione si occuperà, rientrano l'analisi dei dati e l'analisi Costo-Volume-Profitto (CVP).

2.1.1 L'analisi dei dati

Definizione

L'analisi dei dati è un processo attraverso il quale i dati grezzi vengono convertiti in informazioni utili, permettendo ai manager di valutare l'andamento aziendale, prendere decisioni strategiche e monitorare i risultati ottenuti.

Applicazione dell'analisi dei dati

L'applicazione offrirà un'analisi dettagliata delle località per un anno specifico, offrendo una panoramica generale delle loro performance. Questo processo permetterà di identificare i prodotti più richiesti, i periodi di picco nelle vendite, le tendenze stagionali e le categorie di prodotti più redditizie. Sarà inoltre possibile individuare i negozi con i migliori risultati economici, monitorare l'andamento delle vendite nel corso dell'anno e confrontare le performance tra anni diversi.

Vantaggio dell'analisi dei dati

un monitoraggio continuo delle attività permette di comprendere cosa funziona e cosa no. Questo aiuta a:

- ridurre gli sprechi: identificare aree di inefficienza e ottimizzare le risorse.
- contenere i costi: migliorare la gestione dei costi e ottimizzare i costi.
- migliorare la capacità previsionale: effettuare previsioni più accurate sulla domanda, facilitando l'ottimizzazione degli acquisti e garantire un adeguato rifornimento dei prodotti sugli scaffali, soprattutto durante i periodi di picco.

- aumentare l'efficacia delle decisioni strategiche: valutare se sia più vantaggiosa la chiusura di un punto vendita per reinvestire altrove e creare promozioni mirate per stimolare gli acquisti attraverso strategie di marketing.

2.1.2 L'analisi CVP

I manager solitamente sono preoccupati dell'impatto che determinate decisioni possono avere sul profitto aziendale.

Pertanto, nella seconda parte, l'applicazione consentirà agli utenti di eseguire un'analisi CVP (Cost-Volume-Profit), uno strumento utile che permette di simulare diversi scenari aziendali e aiutare i manager a comprendere la relazione esistente tra costi, volume e profitto nelle loro operazioni.

Per capire l'importanza e i limiti di questo strumento previsionale, è necessario comprendere alcuni concetti chiave.

Concetti Chiave dell'analisi CVP

1. Classificazione dei costi:

- Costi variabili (CV): aumentano in proporzione al volume di produzione
- Costi fissi (CF): rimangono invariati, entro certi limiti, al variare della quantità prodotta.

2. Margine di Contribuzione (MDC):

Calcolato come la differenza tra ricavi e costi variabili, rappresenta l'ammontare disponibile per coprire i costi fissi e generare utili nel periodo considerato.

Di conseguenza, il MDC guida la strategia commerciale di un'azienda, fornendo informazioni utili per selezionare i prodotti su cui puntare maggiormente.

3. Margine di Sicurezza (MDS):

Calcolato come la differenza tra vendite previste (o effettive) e vendite corrispondenti al punto di pareggio, fornisce informazioni utili per comprendere di quanto possano calare le vendite prima che si iniziano a registrare delle perdite.

Più alto è il MDS, minore è il rischio di non pareggiare.

4. Punto di Pareggio (BEP):

rappresenta il livello di vendite in cui i costi totali sostenuti sono equivalenti ai ricavi totali ottenuti. In questa situazione, l'azienda non realizza né profitti né perdite.

Il BEP può essere calcolato sia in termini di unità che in termini monetari utilizzando le seguenti formule:

$$BEP(\$) = \frac{CF + TARGET PROFIT}{WACM \text{ ratio}} \qquad BEP(unit) = \frac{CF + TARGET PROFIT}{WACM \text{ unit}}$$

Dove:

Il **profit target** ovvero il livello di profitto desiderato che l'azienda intende raggiungere è pari a zero. Tuttavia, se il manager desidera determinare la quantità ideale per ottenere un certo reddito, allora tale parametro sarà diverso da zero.

Le aziende multi-prodotto, invece di utilizzare il margine di contribuzione per unità al denominatore, impiegano:

- il **margine di contribuzione medio ponderato (WACM ratio)**, calcolato dividendo il margine di contribuzione totale per le vendite totali.
- il **margine di contribuzione medio ponderato per unità (WACM unit)**, calcolato moltiplicando il margine di contribuzione unitario di ogni prodotto per la sua quota di vendite totali. I margini di contribuzione unitari ponderati risultanti per tutti i prodotti vengono poi sommati.

L'applicazione dell'analisi CVP

il software permetterà all'utente di selezionare e inserire i dati necessari per poter avviare le formule per rispondere a domande come:

- Qual è il ricavo ideale da conseguire per raggiungere il pareggio (in cui il ricavo totale è uguale al costo totale, generando un profitto pari a zero) o un determinato profitto?
- Qual è il ricavo ideale per ciascun prodotto?
- Di quanto possono calare i ricavi prima di andare in perdita?
- Che impatto hanno le modifiche apportate ad alcune variabili sul reddito operativo?

Per l'ultima domanda la formula che verrà applicata è la seguente:

$$\Delta RO = MDC_{atteso} - MDC_{attuale} - \Delta CF$$

Limitazioni dell'analisi CVP

Il modello descritto si basa su numerose ipotesi che ne rappresentano i limiti:

- **Distinzione tra costi fissi e variabili:** Alcuni costi possono essere misti, come quelli di manutenzione e ammortamento.
- **Prezzo unitario di vendita invariato:** In realtà, la vendita di unità supplementari può comportare una riduzione del prezzo.
- **Costi variabili per unità costanti:** Questi possono variare a causa di cambiamenti nelle condizioni di mercato, economie di scala, sconti e inflazione.
- **Mix di prodotti costante:** Il mix di vendita, ovvero la proporzione delle vendite di un prodotto rispetto alle vendite totali, può cambiare in risposta a variazioni nelle preferenze dei clienti o a fattori stagionali.
- **le unità prodotte equivalenti alle unità vendute:** In realtà le unità prodotte potrebbero non essere uguali a quelle vendute a causa del cambiamento della domanda

Queste limitazioni indicano che l'analisi del CVP risulta maggiormente appropriata per il processo decisionale a breve termine, in contesti in cui i fattori che influenzano costi e ricavi sono relativamente stabili e prevedibili.

Questo strumento si rivela particolarmente utile per aziende con un prodotto semplice e omogeneo, dove il mix di prodotti e il livello delle scorte non rivestono un ruolo significativo.

Nonostante tali osservazioni, l'analisi CVP, se usato con cautela, rimane un valido strumento per la pianificazione e il controllo.

2.2. L'importanza dell'autenticazione nelle PMI

Le piccole e medie imprese (PMI) spesso detengono informazioni critiche il cui furto o compromissione potrebbe pregiudicare il loro vantaggio competitivo.

In questo contesto, l'autenticazione si configura come un elemento fondamentale della strategia aziendale.

Definizione dell'Autenticazione

il **National Institute of Standards and Technology (NIST)** definisce l'autenticazione come un processo di verifica dell'identità di un utente o di un dispositivo, necessario per consentire l'accesso alle risorse all'interno di un sistema informativo.

Normativa di Riferimento

In Italia, il **Regolamento Generale sulla Protezione dei Dati (GDPR)** stabilisce requisiti specifici per la protezione dei dati personali, inclusa l'autenticazione.

L'articolo 32 del GDPR richiede che le aziende implementino misure tecniche e organizzative adeguate per garantire un livello di sicurezza proporzionato al rischio, inclusa l'autenticazione forte per proteggere l'accesso ai dati sensibili.

Implementazione dell'autenticazione

Il software prevede una fase di autenticazione attraverso il quale solo chi ha le credenziali di accesso (username e password) è autorizzato a consultare i dati presenti nel database.

3. DESCRIZIONE DEL DATA-SET UTILIZZATO

Il **data-set aziendale** utilizzato, reperito dal sito **Kaggle.com** con una licenza di utilizzo non commerciale, consiste in quattro file CSV contenenti dati relativi a una catena di negozi di giocattoli con sede in Messico, comprendo il periodo da gennaio 2022 a settembre 2023.

Prima dell'importazione nel database SQL, sono state effettuate diverse modifiche ai dati forniti nel file Excel per garantire l'integrità e l'affidabilità delle informazioni.

In particolare:

- Il formato del campo **data** è stato adattato per assicurare la compatibilità con il formato richiesto dal database
- Sono stati integrati alcuni dati mancanti nella tabella **inventory**, poiché alcuni negozi non disponevano di informazioni su determinati prodotti. Si è pertanto assunto che lo stock di tali prodotti fosse pari a zero.

Il data-set su cui si basa il software è disponibile nel progetto come file “maventoystore.sql”, situato all'interno della cartella “**database**”.

3.1 La struttura dei Dati

Il data-set è composto da quattro tabelle principali:

- **Products:**

- Composta da 35 righe, questa tabella rappresenta i prodotti offerti dai negozi.
- Contiene l'identificativo del prodotto, il nome, il costo, il prezzo e la categoria di appartenenza

- **Stores:**

- Composta da 50 righe, questa tabella rappresenta la catena dei negozi presenti in Messico.
- Contiene l'identificativo del negozio, il nome corrispondente e la città in cui si trova. Il campo data si riferisce all'apertura dello store.

- **Sales:**

- Composta da circa 344 miliardi di righe, questa tabella rappresenta i prodotti venduti dai negozi nel periodo compreso tra gennaio 2022 e luglio 2023.
- Contiene l'identificativo dell'acquisto relativo ad un determinato prodotto, individuato dal Product_id, in uno specifico negozio, contraddistinto dal Store_id.

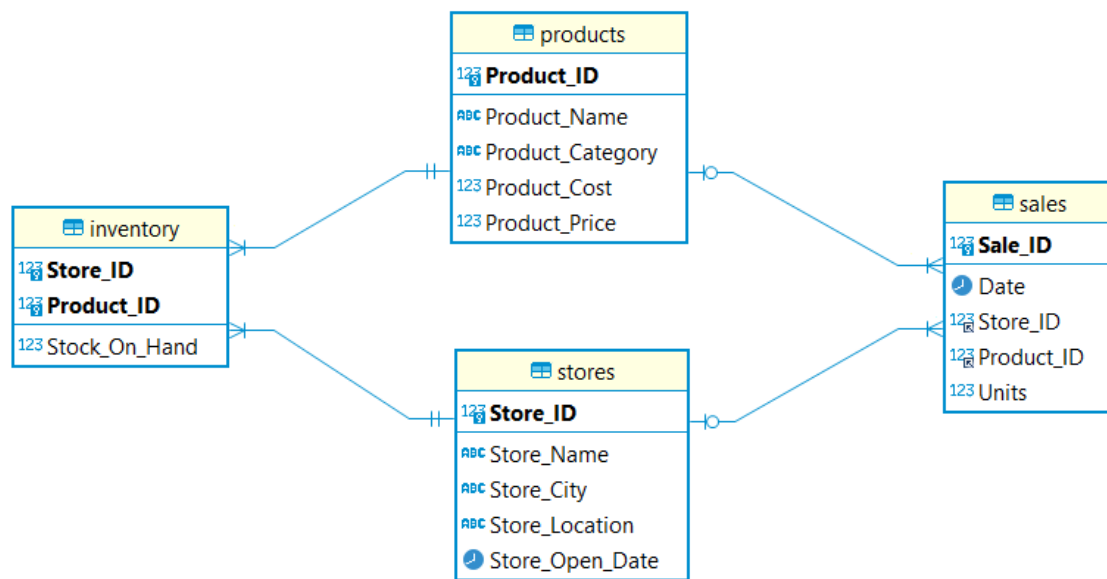
Il campo data indica il momento in cui è stata effettuata la transazione

- **Inventory:**

- Composta da 1740 righe, questa tabella rappresenta lo stock disponibile nei vari negozi.
- registra i diversi livelli di inventario disponibili in ogni negozio, identificato dallo Store_id, per il prodotto corrispondente al codice identificativo Product_id.

3.2 Diagramma ER

Il diagramma ER del dataset utilizzato è:

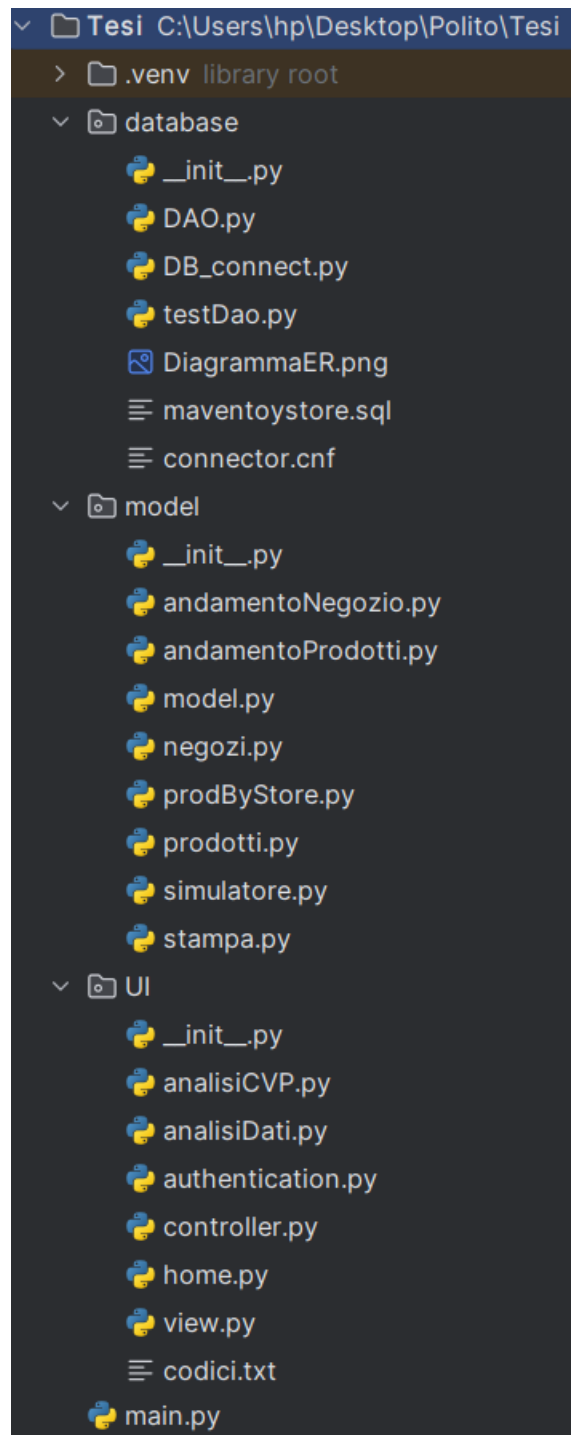


4. DESCRIZIONE AD ALTO LIVELLO DELLE STRUTTURE DATI E DEGLI ALGORITMI COINVOLTI

4.1 Struttura del progetto

L'applicazione è stata sviluppata in linguaggio di programmazione Python, seguendo i pattern architetturali MVC (Model View Controller) e DAO (Data Access Object), che favoriscono una chiara separazione delle responsabilità e una maggiore manutenibilità del codice.

L'interfaccia è stata implementata tramite la libreria **Flet** di Python.



La struttura del software è suddivisa nel seguente modo:

- **Main:** responsabile dell'avvio dell'applicazione.
- **Package UI** contiene:
 - La **classe view** che è incaricata di creare l'interfaccia grafica e gestisce il layout delle pagine.
 - La **classe Controller** che raccoglie tutti gli eventi, ovvero le azioni dell'utente, e le gestisce inoltrandoli al model, che elabora le richieste.
 - Le **classi** rappresentano una vista specifica dell'applicazione.
 - Il file **"codici.txt"** dove sono contenuti le credenziali di accesso
- **Package database** contiene:
 - La **classe DBConnect** che permette di avere un canale di comunicazione con il database.
 - La **classe DAO** che contiene i metodi necessari per ottenere informazioni dal data-set attraverso opportune interrogazioni (query) e inviarli successivamente al model.
 - La **classe TestDao**, sebbene non abbia un ruolo diretto nell'economia del progetto, questa classe ha fornito supporto durante la fase di sviluppo, consentendo di testare le funzionalità del DAO e assicurare che le interrogazioni restituissero risultati corretti.
 - Il **Data-set "maventoystore.sql"** che viene utilizzato nel progetto
 - Il file **connector.cnf** che contiene i dati necessari per accedere al database

- **Package Model** contiene:

- La **classe Model** che funge da passacarte tra il DAO e il Controller. Essa fornisce i metodi applicativi necessari per eseguire operazioni sui dati.
- Le **classi** che definiscono oggetti e metodi con cui abbiamo a che fare nel programma.

4.2 Algoritmi utilizzati nel progetto

Definizione di Algoritmo

Un **algoritmo** è definito come una successione di istruzioni o passi che definiscono le operazioni da eseguire sui dati per ottenere i risultati desiderati.

All'interno del package Model, si trovano i metodi necessari per avviare gli algoritmi.

4.2.1. Algoritmi ricorsivi

Gli **algoritmi ricorsivi** sono caratterizzati dalla loro capacità di richiamare sé stessi, generando una sequenza di chiamate che ha termine al verificarsi di una condizione particolare, nota come **condizione di terminazione**.

Tali algoritmi consentono di semplificare la risoluzione di problemi complessi, scomponendoli in sottoproblemi più gestibili, facilitando la comprensione, la manutenzione e la riusabilità del codice.

1. Metodo “ricorsione1”

All'interno di un ciclo for, vengono esaminati i negozi e viene passata la lista dei profitti ottenuti alla funzione **ricorsione1**, la quale restituisce il profitto massimo associato a ciascun negozio e il corrispondente indice. Qualora questo profitto risulti essere il massimo attualmente registrato, la variabile max_profit verrà aggiornata di conseguenza. Il ciclo continua fino a quando tutti i negozi sono stati esaminati.

Parametri della funzione

- lst: Lista contenente i margini di contribuzione di ogni prodotto.
- index: Indice che tiene traccia della posizione corrente nella lista durante la ricorsione.
- highest: Valore massimo attualmente indentificato nella lista.
- i: Indice del valore massimo.

funzionamento della funzione

La funzione verifica se il valore corrente è maggiore o uguale a highest. In caso affermativo, il valore di highest viene aggiornato di conseguenza.

La funzione richiamerà sé stessa incrementando index di 1, permettendo di esaminare il prossimo elemento della lista. Questo processo continua fino all'esaurimenti della lista, al termine del quale la funzione restituisce il valore massimo corrente e l'indice corrispondente

```

class Model:
    def get_prodotto_most_redditizio(self, loc, yy):
        """ metodo per ricavare il negozio che presenta il prodotto più redditizio e la categoria a cui appartiene """

        # Inizializzo le variabili contenenti i risultati
        max_profit = 0
        best_solution = 0
        p, n = self.get_andamento(loc, yy)
        for t, i in n.items():
            # Calcolo i profitti tramite ricorsione
            profit, index = self.ricorsione1(i[3])
            # controllo se il profitto sia >= massimo profitto trovato
            if profit >= max_profit:
                max_profit = profit # Aggiorno il profitto Massimo
                prod = i[0][index] # ricavo il prodotto corrispondente
                category = self.get_prodotti(prod).product_category # Ricavo la categoria del prodotto
                # Aggiorno la migliore soluzione
                best_solution = (t, f"Product:\n{prod}\n[{category}]", f"Profit:\n${max_profit:1.2f}")

        return best_solution

    def ricorsione1(self, lst, index=0, highest=None, i=None):
        """ Funzione ricorsiva per trovare gli indici e i valori max in una lista """

        # Condizione di uscita dalla ricorsione
        if index >= len(lst):
            return highest, i

        if highest is None or lst[index] >= highest:
            highest = lst[index] # Aggiorno il valore max
            i = index # Aggiorno l'indice

        # Richiamo la ricorsione sul prossimo indice
        return self.ricorsione1(lst, index + 1, highest, i)

```

2. Metodo “ricorsione2”

La funzione ricorsione2 è stata progettata per determinare il ricavo ideale di ciascun prodotto e ricavare il mdc medio ponderato necessario per poter calcolare il punto di pareggio in unità.

Parametri della funzione

- qty: Lista delle quantità vendute di ciascun prodotto.
- rr: Lista dei ricavi ottenuti da ciascun prodotto.
- mdc: Lista del mdc di ciascun prodotto.
- totq: Somma totale delle quantità vendute.
- totr: Somma totale dei ricavi generati.
- index: Indice che tiene traccia della posizione corrente nella lista durante la ricorsione.
- wamdc: Margine di contribuzione medio ponderato accumulato fino a quel punto.
- mix: Lista che raccoglie i ricavi ideali per ciascun prodotto.

Funzionamento della funzione

Per ciascun prodotto, viene calcolato la percentuale della quantità di quel prodotto rispetto al totale delle quantità vendute. Tale percentuale viene poi moltiplicato per il margine di contribuzione unitario per ottenere il margine di contribuzione medio ponderato.

Questo valore viene accumulato fino al raggiungimento della condizione di terminazione della ricorsione.

Successivamente si calcolerà la percentuale del ricavo per il prodotto attuale rispetto al totale dei ricavi. Tale percentuale viene moltiplicata per il punto di pareggio in dollari per ottenere il ricavo ideale per il prodotto in esame, che viene poi aggiunto alla lista mix.

Se mix è None, viene inizializzato come una lista vuota.

La condizione di uscita dalla ricorsione si verifica quando index raggiunge o superato la lunghezza della lista qty indicando che tutti i prodotti sono stati esaminati.

In questo caso, la funzione restituisce il valore accumulato di wamdc e la lista mix contenente i ricavi ideali per ciascun prodotto.

In caso contrario, la funzione richiamerà sé stessa incrementando index di 1 e passando i valori aggiornati di wadmc e mix.

```
class Model:
    def calcolo_mdc(self, shop, yy, mm, cf, target):
        """ Calcola il MDC per il negozio specificato in base ai parametri forniti """

        # ricavo i dati per il mese specificato
        name_prod, qty, revenue, mdc = self.andamento1(shop, yy).get(mm)

        # calcola il RO sottraendo i costi fissi
        self.ro = sum(mdc) - cf

        # calcola il MDC medio
        wamdc = sum(mdc) / sum(revenue)
        # calcolo il punto di pareggio in $
        self.rbep = (cf + target) / wamdc

        tot_qty = sum(qty)
        tot_revenue = sum(revenue)
        # chiama la funzione ricorsiva per calcoli aggiuntivi
        wamdc, mixx = self.ricorsione2(qty, revenue, mdc, tot_qty, tot_revenue)

        # calcolo il punto di pareggio in unita
        self.qbep = (cf + target) / wamdc

        # calcolo il margine di sicurezza
        self.mds = sum(revenue) - self.rbep

        # memorizzo i risultati
        self.ris1 = (name_prod, qty, revenue, mixx)
```

```
def ricorsione2(self, qty, rr, mdc, totq, totr, index=0, wamdc=0, mix=None):
    """ Funzione ricorsiva per calcolare variabili relative a vendite e margini per ogni prodotto """

    if mix is None:
        mix = []

    # Condizione di uscita dalla ricorsione
    if index >= len(qty):
        return wamdc, mix

    # Calcolo il MDC unitario
    u = mdc[index] / qty[index]
    mixu = qty[index] / totq # <-- sales mix in % unit
    wamdc += mixu * u

    # per ottenere il ricavo da ottenere per ogni prodotto
    mixr = rr[index] / totr
    rbepi = mixr * self.rbep
    mix.append(rbepi)

    return self.ricorsione2(qty, rr, mdc, totq, totr, index + 1, wamdc=wamdc, mix=mix)
```

3. Metodo “ricorsione3”

La funzione ricorsione3 è stata progettata per restituire una lista di rette.

Parametri della funzione

- punti: Lista di tuple, in cui ogni tupla rappresenta un punto (x, y) nel piano cartesiano.
- x1: Array di valori sull'asse x.
- index: Indice che tiene traccia della posizione corrente nella lista dei punti durante l'esecuzione della ricorsione.

Funzionamento della funzione

La funzione seleziona due punti dalla lista punti e invoca un metodo ausiliario, passando i punti a e b insieme all'array x1. Questo metodo calcola l'equazione della retta che passa attraverso i punti, restituendo i valori corrispondenti per l'asse y.

La chiamata ricorsiva continua fino a quando tutti i punti sono stati elaborati. Durante ogni chiamata, l'indice viene incrementato di 1, consentendo di elaborare il prossimo paio di punti. La retta calcolata viene aggiunta alla lista di risultati, che verrà restituita al termine della ricorsione.

```
class Model:
    def get_retta(self, cf):
        """ Calcolo la retta per rappresentare graficamente i risultati in base ai costi fissi """

        # Calcolo la quantità totale venduta
        q = sum(self.get_ris1()[1])

        # Ricavo i valori necessari per la retta
        qbep, rbep, mds, ro = self.get_bep_e_mds()

        # Creo i valori per l'asse x
        x1 = np.linspace(start=0, q + qbep)

        # Definisco i punti per la retta
        pt = [(0, 0), (qbep, rbep), (0, cf)]

        return self.ricorsione3(pt, x1)

    def ricorsione3(self, punti, x1, index=0):
        """ Funzione ricorsiva per calcolare le rette tra punti dati
        e restituire una lista di rette """

        # Condizione di uscita dalla ricorsione
        if index >= len(punti)-1:
            return [] # se non ci sono più punti da elaborare, restituisce una lista vuota

        a = punti[index]
        b = punti[index + 1]

        # Ricavo equazione della retta passante tra i due punti dati
        retta = self.creazione_retta(a, b, x1)

        # Restituisce la retta e continua la ricorsione
        return [retta] + self.ricorsione3(punti, x1, index + 1)
```

4.2.2. Simulazione

La **simulazione** è un processo di modellazione di fenomeni reali attraverso un insieme di formule matematiche che permette di analizzare e prevedere i risultati, facilitando decisioni informate e aumentando la competitività del negozio.

Nello specifico, la simulazione in questione utilizza le formule dell'analisi del **Cost-Volume-Profit (CVP)**, uno strumento che permette all'utente di osservare come varia il reddito operativo del negozio X nel mese di interesse al variare di determinate variabili, utilizzando una coda per gestire gli eventi e garantire un'esecuzione efficiente della simulazione.

Innanzitutto, si procede formulando alcune considerazioni preliminari:

- **Mix delle vendite costante:** Si suppone che la distribuzione delle vendite tra i vari prodotti rimanga invariata.
- **Percentuali prestabiliti:** La simulazione opera selezionando da un insieme finito di percentuali predeterminate, utilizzate per variare le variabili al fine di individuare soluzioni alternative.
- **Numero di soluzioni alternative:** Si è stabilito di trovare max 10 soluzioni alternative.

Una volta inserite le variazioni delle variabili e premuto il pulsante "Calcola", verrà avviato il metodo **simulazione**.

Parametri della funzione

- **vqty:** Variabile che rappresenta la variazione percentuale delle quantità vendute.
- **vpr:** Variabile che rappresenta la variazione percentuale del prezzo di vendita.
- **vcf:** Variabile che rappresenta la variazione che subisce il costo fisso.
- **prod:** Dizionario contenente informazioni sui prodotti venduti (nome, prezzo unitario, costo variabile unitario, quantità venduta).
- **Mdc:** Margine di contribuzione calcolato senza considerare le variazioni.

Funzionamento della funzione

La funzione **sim.run()** gestisce il processo di simulazione, elaborando ogni evento presente nella coda fino a quando quest'ultima non risulta vuota. Gli eventi possono essere classificati in tre categorie: **inizio**, **fine** e **nuovo**, e il comportamento della simulazione varia in base al tipo di evento.

Tipo Inizio

Quando viene elaborato un evento di tipo **inizio**, viene invocata la funzione **calcolo_mdc_atteso()** per calcolare il margine di contribuzione atteso. Da questo valore verranno sottratti sia il margine di contribuzione attuale sia la variazione del costo fisso, al fine di ottenere la variazione del reddito operativo.

Successivamente, viene calcolato il tempo di esecuzione per questa procedura, che sarà utilizzato per caricare la coda con un nuovo evento di tipo **fine**.

Tipo Fine

Nell'ambito di un evento di tipo **fine**, viene effettuato un controllo per verificare che il tempo di esecuzione della simulazione non superi i 2 secondi e che le soluzioni alternative richieste

non siano già state identificate. Se la variazione del reddito operativo risultante è positiva, il valore verrà memorizzato in un elenco contenente i risultati ottenuti.

Infine, la coda verrà ricaricata con un nuovo evento di tipo **nuovo**.

Tipo Nuovo

Per un evento di tipo **nuovo**, viene invocata la funzione **genero_percentuali()** per generare nuove percentuali relative alle variabili. Se il tempo di esecuzione per questa procedura è inferiore a 2 secondi e tutte le combinazioni non sono state già provate, si verifica se le nuove percentuali generate sono già presenti nel sistema. In caso contrario, si decrementa il numero di combinazioni possibili e la coda viene ricaricata con un nuovo evento di tipo **inizio**, decrementando contemporaneamente il numero di soluzioni trovate e aggiornando il tempo di esecuzione.

```
class Model:
    def simulazione(self, cf, vqty, vpr, vcf):
        """ Esegue una simulazione basata sui parametri forniti """

        # ricavo i parametri che devono essere passati:
        nameprod, qty, rr, mixx = self.get_ris1()
        mdc = self.ro+cf # Calcolo il MDC attuale
        prod = {} # lista che contiene informazioni riguardanti ai prodotti (Pname, Pr, cv, qty)
        for i in range(len(nameprod)):
            prod.update({nameprod[i]: (self.get_prodotti(nameprod[i]).product_price, self.get_prodotti(nameprod[i]).product_cost, qty[i])})

        # Passo i parametri al simulatore:
        sim = Simulatore(prod, mdc)
        sim.init(vqty, vpr, vcf) # passo i parametri che vengono dal controller
        # avvio la simulazione
        sim.run()
        # memorizzo i risultati della simulazione
        self._ris2 = sim.get_result()

    @usage(1 dynamic) new *
    def get_ris2(self):
        """ Restituisce i risultati della simulazione """
        return self._ris2
```

```
class Simulatore:
    def process_events(self, e):
        """ Processa gli eventi nella coda """

        t = e[0] # tempo dell'evento
        tipo = e[1] # tipo di evento
        n = e[2] # Numero di soluzioni rimanenti

        if tipo == 'inizio':
            # Calcolo la variazione del RO atteso
            self.ris = self.calcolo_mdc_atteso() - self.mdc_attuale - self.vCF

            # Calcolo il tempo richiesto per eseguire questa procedura
            tf = time.perf_counter()
            te = tf-self.ti

            # Aggiunge un nuovo evento alla coda
            heapq.heappush(*args: self.queue, (te, 'fine', n))

        if tipo == 'fine':
            # controllo che le soluzioni alternative non siano state già trovate
            # e che il tempo di esecuzione non sia superiore a 2 secondi
            if n > 0 and t < 2:
                # controllo che RO calcolato sia positivo
                if self.ris > 0:
                    # memorizzo le soluzioni trovate
                    self.ro.update({(self.vQty, self.vPr, self.vCF): self.ris})

            # carico la coda:
            heapq.heappush(*args: self.queue, (t, 'nuovo', n))
```

```
if tipo == 'nuovo':
    # Genero nuove percentuali per le variabili
    trovato = self.genero_percentuali()

    # calcolo il tempo richiesto per la procedura precedente
    tf = time.perf_counter()
    te = tf-self.ti

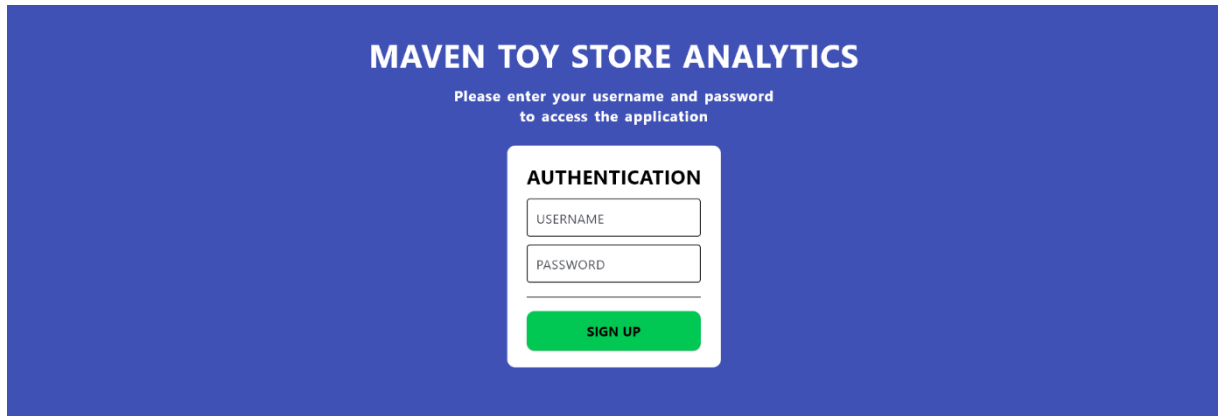
    # controllo che tutte le combinazioni possibili siano state provate
    # e che il tempo di esecuzione non sia superiore a 2 secondi
    if te < 2 and self.num_combinazioni > 0:
        if trovato is True:
            self.num_combinazioni -= 1 # decremento il numero di combinazioni possibili
            # carico la coda aggiornando il numero di soluzioni trovate e il tempo di esecuzione
            heapq.heappush(*args: self.queue, (te, 'inizio', n-1))
        else:
            heapq.heappush(*args: self.queue, (te, 'nuovo', n))
```

5. ALCUNE VIDEATE DELL'APPLICAZIONE REALIZZATA E LINK AL VIDEO DIMOSTRATIVO DEL SOFTWARE

5.1 Videata all'apertura dell'applicazione

Il software è stato progettato affinché solo gli utenti autorizzati possano accedere e consultare i dati. Questo approccio è fondamentale per proteggere le informazioni.

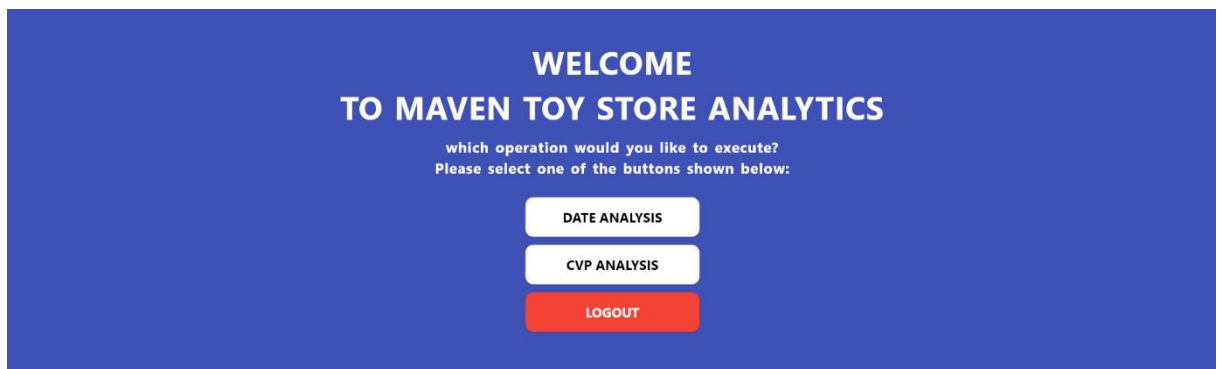
Per questo motivo una volta avviato il programma, la videata che apparirà all'utente mostrerà i campi per l'inserimento di username e password affinché il programma possa verificarne l'accessibilità.



Se i dati inseriti non sono corretti, verrà visualizzato un messaggio di errore, nel caso contrario, se le credenziali fornite sono corrette, l'utente verrà reindirizzato a una pagina successiva.

In questa nuova schermata, saranno disponibili due pulsanti che permetteranno all'utente di selezionare l'operazione di suo interesse.

Inoltre, sarà presente un pulsante di logout, che consentirà all'utente di disconnettersi.



5.2 Videata dei risultati

Pagina “data analysis”

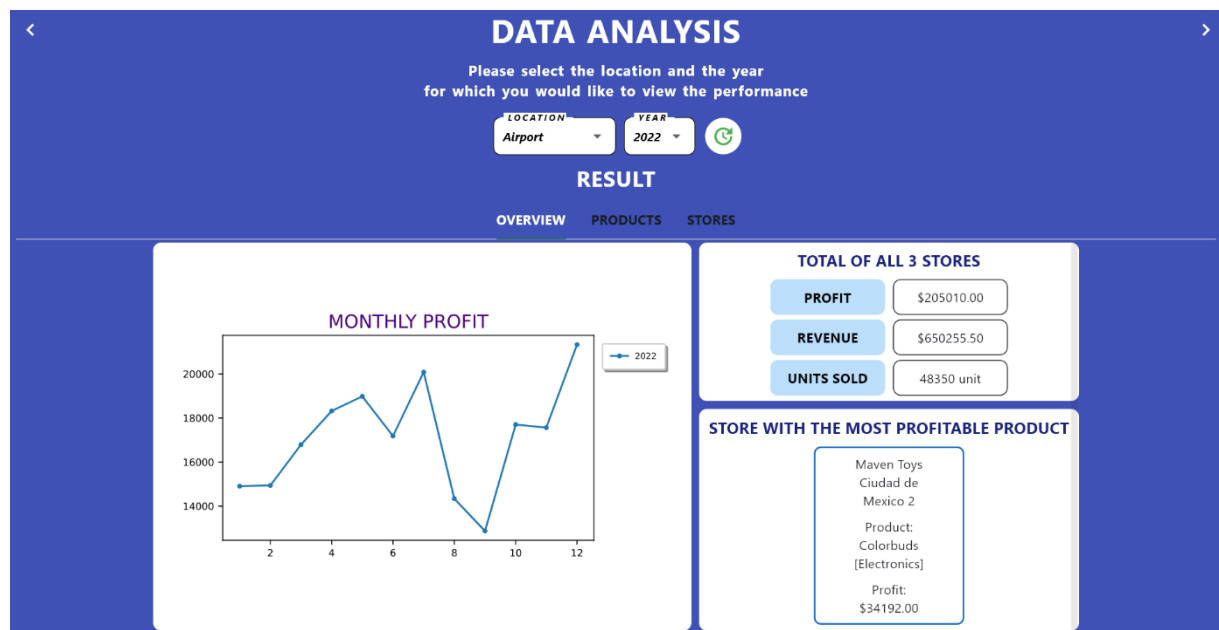
La pagina “**data analysis**” presenta 2 menu a tendina che consentono di selezionare la località e l’anno di interesse.

Cliccando sul pulsante **Aggiorna**, l’utente potrà visualizzare i risultati in tre sezioni distinte.

1. Tab “overview”

La prima tab, “**overview**”, fornisce un’analisi generale della location selezionata. Presenta:

- un grafico che illustra l’andamento del profitto nella località selezionata.
- Il numero di negozi presenti nella località in esame, con l’indicazione totale delle quantità vendute, del ricavo e del profitto realizzato nell’anno specificato.
- il negozio che offre il prodotto più redditizio, specificando la categoria a cui il prodotto appartiene e il profitto generato

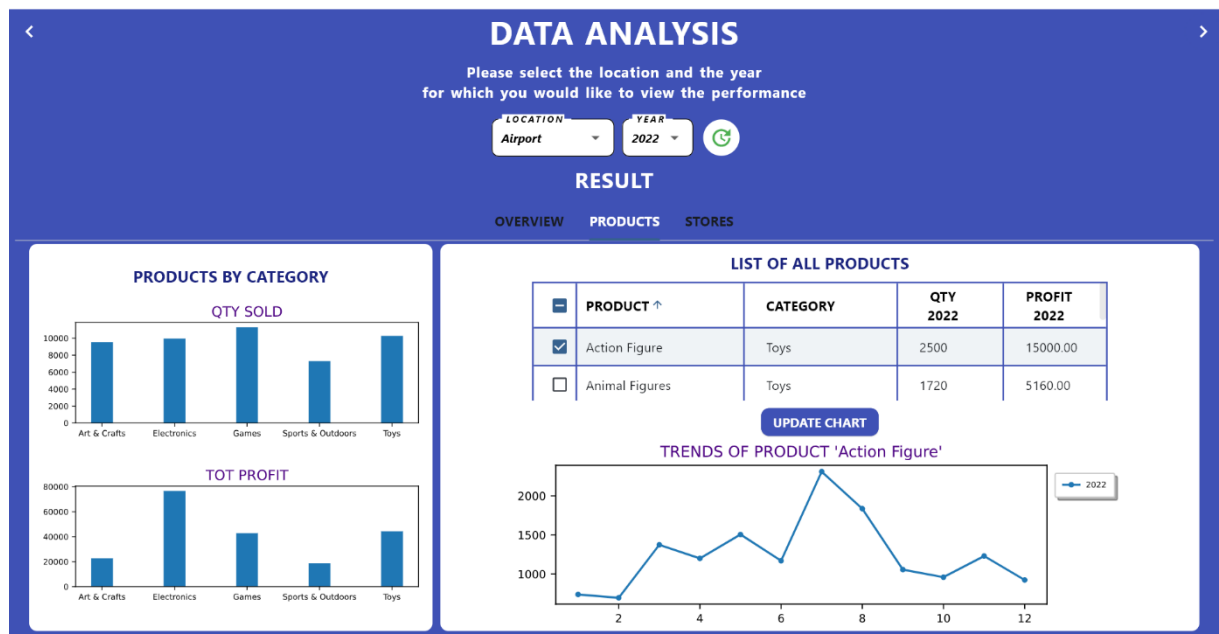


2. Tab “product”

La seconda tab, “**product**”, fornisce una analisi dettagliata dei prodotti disponibili nei negozi. Include:

- un grafico a barre che permette di identificare quale categoria di prodotti risulta più redditizia
- una tabella che elenca tutti i prodotti offerti dai negozi.

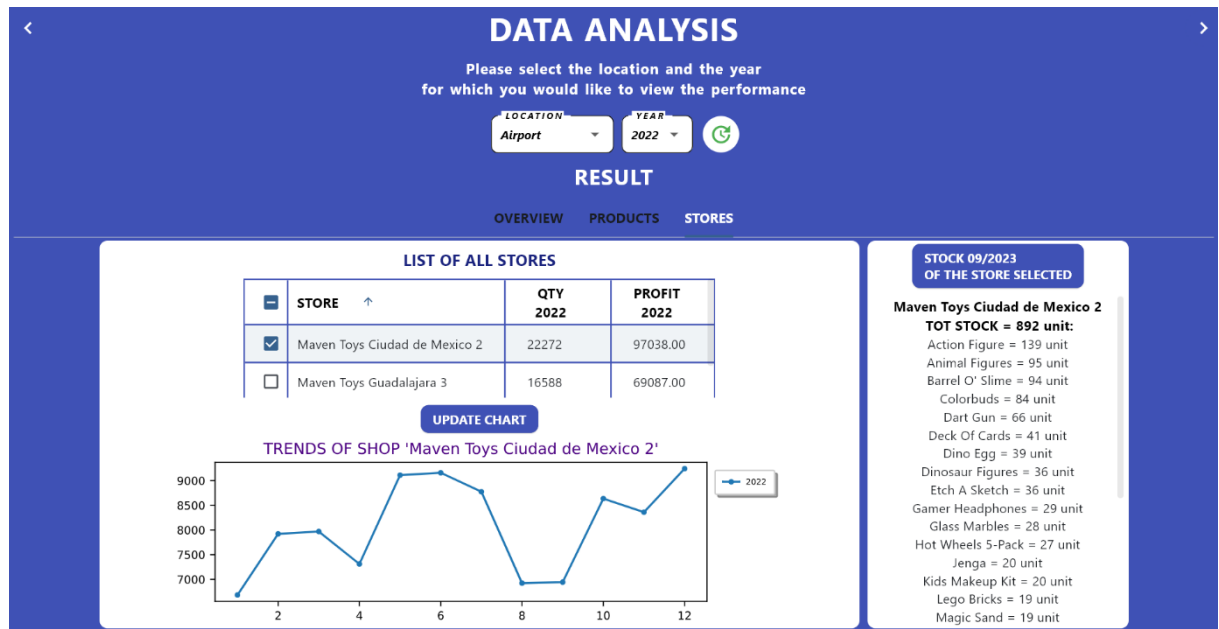
Selezionando un prodotto specifico, l’utente potrà visualizzare l’andamento mensile delle vendite relative a quel prodotto.



3. Tab “store”

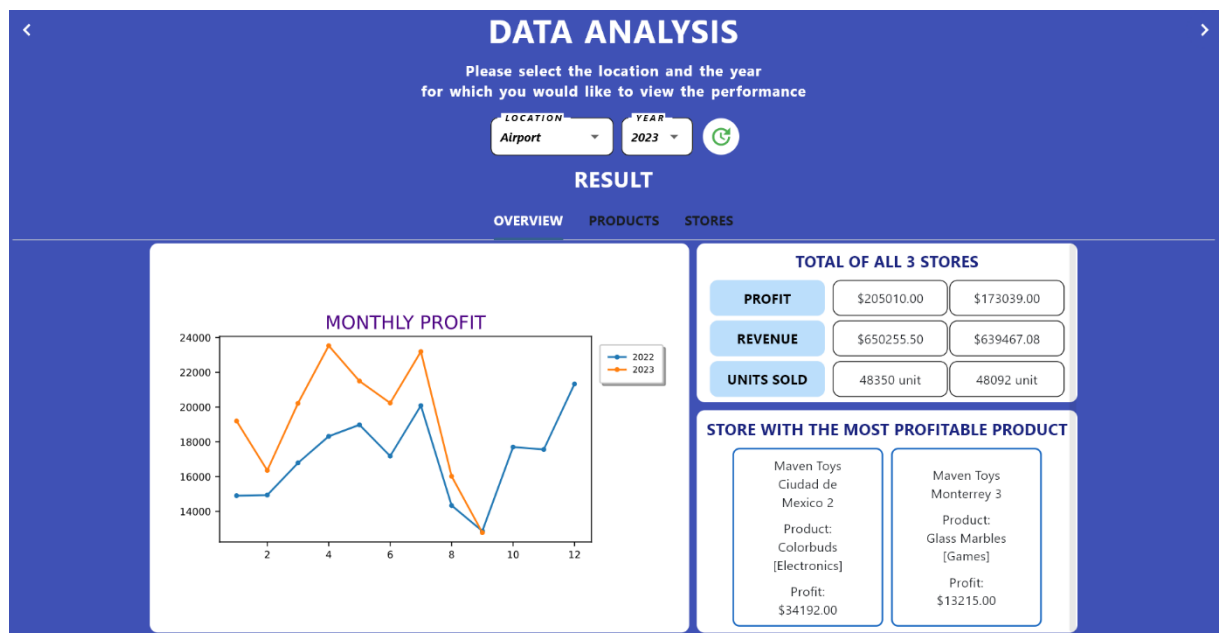
La terza tab, “store”, fornisce un’analisi dettagliata dei punti vendita presenti nella località selezionata. Include:

- una tabella che elenca tutti i negozi operanti nella località. Selezionando un punto vendita, l’utente potrà visualizzare l’andamento mensile delle vendite e lo stock disponibile a fine settembre 2023.



Inoltre, gli utenti hanno la possibilità di ordinare i dati presenti nelle tabelle selezionando la colonna desiderata, consentendo al programma di eseguire l’ordinamento in base alla scelta effettuata.

Nel caso in cui l'utente modifichi solo l'anno, il software terrà conto di questa variazione, permettendo confronti tra diversi anni.

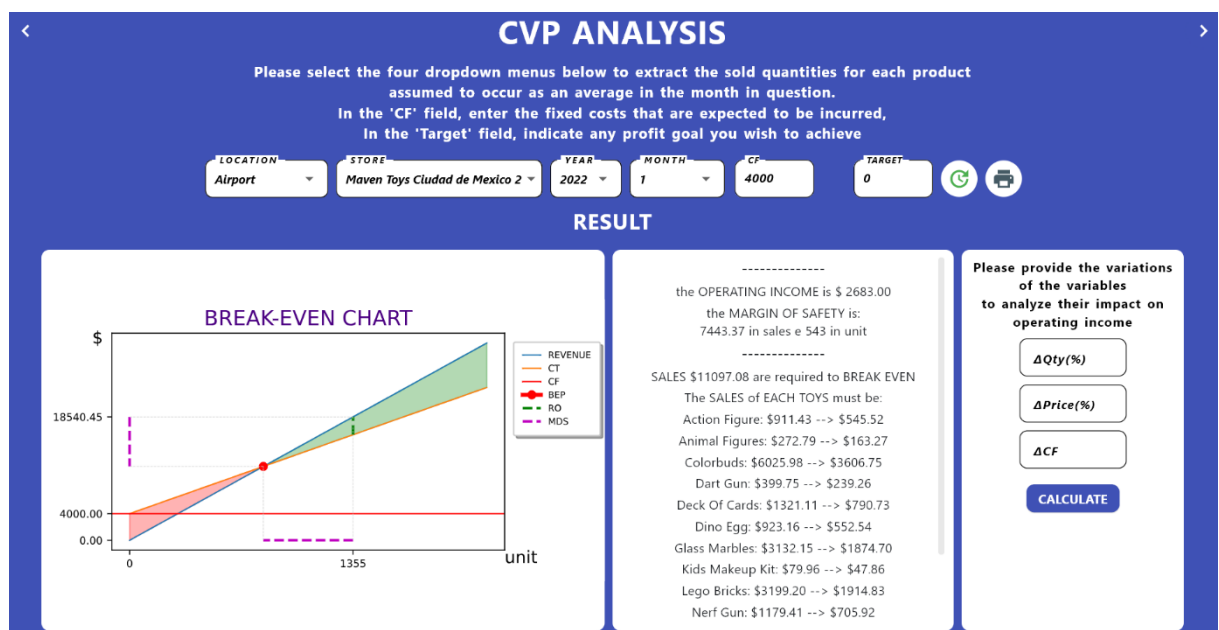


Pagina "analysis CVP"

La pagina "analysis CVP" mette a disposizione quattro menu a tendina che consentono all'utente di indicare le quantità di vendita che presuppone di avere per ogni prodotto anche nel mese in esame.

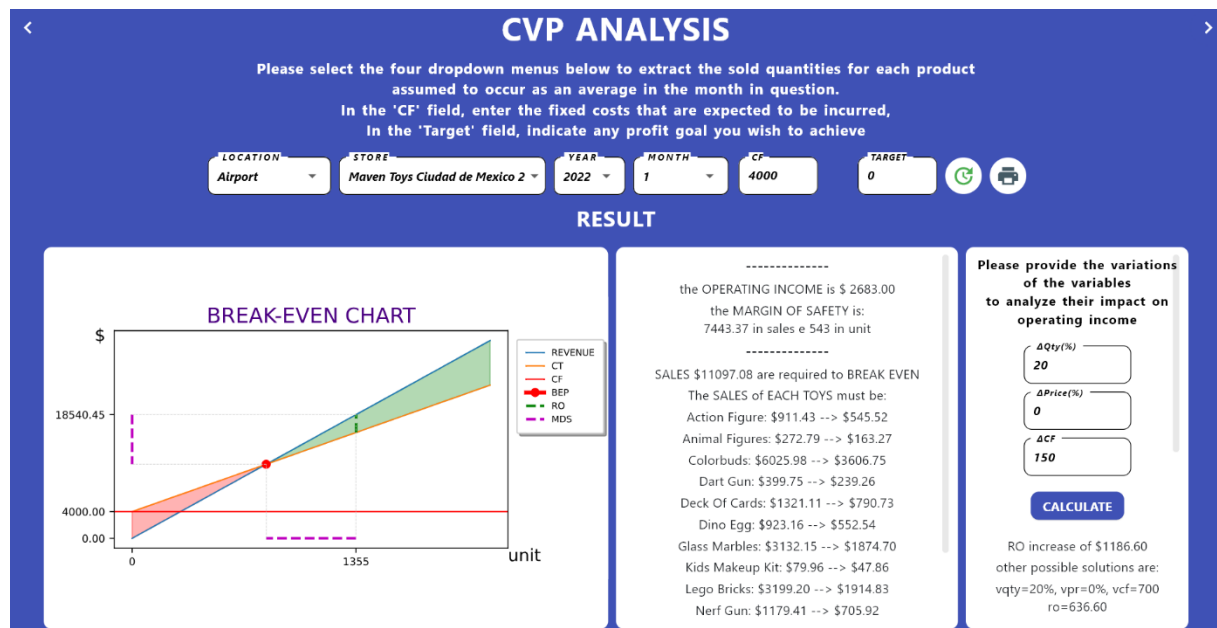
Inoltre sono presenti due campi di testo, in cui l'utente potrà inserire i costi fissi che prevede di sostenere, mentre nel secondo può indicare l'eventuale obiettivo di profitto che vuole realizzare. Questi valori saranno utilizzati per implementare le formule presenti nella sezione 2.1.2.

L'analisi CVP avrà inizio solo se i dati verranno inseriti nel formato corretto altrimenti verrà mostrato un messaggio di errore.



L'utente potrà valutare quale effetto abbiano le modifiche a determinate variabili sul reddito operativo, inserendo le variazioni pertinenti.

Tali dati sono necessari per applicare le formule riportate nella sezione 2.1.2.



Ad esempio, si ipotizza che un incremento del budget per il marketing di \$150 possa generare un aumento delle vendite del 20%. Si Osserva che tale variazione comporterebbe un aumento dell'reddito operativo di \$1186,60. Pertanto, questa variazione potrebbe essere considerata vantaggiosa da implementare.

L'applicazione offrirà anche ulteriori strategie di variazioni che possano generare un aumento del reddito operativo.

Infine, sarà possibile stampare i risultati in un file PDF, cliccando sul pulsante di stampa. Tale pulsante richiederà all'utente di inserire il nome con cui desidera salvare il documento.

CVP ANALYSIS

Please select the four dropdown menus below to extract the sold quantities for each product assumed to occur as an average in the month in question.
In the 'CF' field, enter the fixed costs that are expected to be incurred,
In the 'Target' field, indicate any profit goal you wish to achieve

LOCATION: Airport | STORE: Maven Toys Ciudad de Mexico 2 | YEAR: 2022 | MONTH: 1 | CF: 4000 | TARGET: 0

RESULT

BREAK-EVEN CHART

The chart shows the Break-Even Point (BEP) at 1355 units. The Y-axis represents dollars (\$), ranging from 0.00 to 18540.45. The X-axis represents units, ranging from 0 to 1355. The chart includes lines for REVENUE (blue), CT (orange), CF (red), RO (green), and MDS (purple).

the OPERATING INCOME is \$ 2683.00
the MARGIN OF SAFETY is:
7443.37 in sales e 543 in unit

SALES \$11097.08 are required to BREAK EVEN
The SALES of EACH TOYS must be:
Action Figure: \$911.43 --> \$545.52
Animal Figures: \$272.79 --> \$163.27
Colorbuds: \$6025.98 --> \$3606.75
Dart Gun: \$399.75 --> \$239.26
Deck Of Cards: \$1321.11 --> \$790.73
Dino Egg: \$923.16 --> \$552.54
Glass Marbles: \$3132.15 --> \$1874.70
Kids Makeup Kit: \$79.96 --> \$47.86
Lego Bricks: \$3199.20 --> \$1914.83
Nerf Gun: \$1179.41 --> \$705.92

Please provide the variations of the variables to analyze their impact on operating income

$\Delta Qty(\%)$: 20
 $\Delta Price(\%)$: 0
 ΔCF : 150

CALCULATE

RO increase of \$1186.60
other possible solutions are:
vqty=20%, vpr=0%, vcf=700
ro=636.60

Please indicate the name you wish to give to the file

CONFIRM **DELETE**

Successivamente, dopo aver dato conferma, il programma comunicherà il successo dell'operazione.

Di seguito viene presentato un esempio dei risultati ottenuti, salvati in formato PDF.

CVP ANALYSIS RESULT

(FIX COST= \$4000.0, TARGET= \$0.0, QUANTITY= specified in the "QTY" column of the table)

the SALES is \$18540.45

the OPERATING INCOME is \$2683.00

the MARGIN OF SAFETY is 7443.37 in sales

SALES \$11097.08 are required to BREAK EVEN

The SALES of EACH TOYS must be:

PRODUCT	QTY	REVENUE	IDEAL REVENUE
Action Figure	57 unit	\$911.43	\$545.52
Animal Figures	21 unit	\$272.79	\$163.27
Colorbuds	402 unit	\$6025.98	\$3606.75
Dart Gun	25 unit	\$399.75	\$239.26
Deck Of Cards	189 unit	\$1321.11	\$790.73
Dino Egg	84 unit	\$923.16	\$552.54
Glass Marbles	285 unit	\$3132.15	\$1874.70
Kids Makeup Kit	4 unit	\$79.96	\$47.86
Lego Bricks	80 unit	\$3199.20	\$1914.83
Nerf Gun	59 unit	\$1179.41	\$705.92
PlayDoh Can	47 unit	\$140.53	\$84.11
PlayDoh Toolkit	40 unit	\$199.60	\$119.47
Rubik's Cube	18 unit	\$359.82	\$215.36
Splash Balls	44 unit	\$395.56	\$236.76

The following analysis examines the variation in operating income resulting from specific modifications to variables such as units sold, price and fixed costs

VarQty	VarPrice	VarCF	VarRO
20%	0%	700	\$636.60
20%	0%	350	\$986.60
20%	0%	300	\$1036.60
20%	0%	250	\$1086.60
20%	0%	200	\$1136.60
20%	0%	150	\$1186.60
20%	0%	100	\$1236.60
20%	0%	-200	\$1536.60
20%	0%	-600	\$1936.60
20%	0%	-750	\$2086.60

5.3 Link YouTube

Il video di dimostrazione di utilizzo del software è disponibile al seguente link di YouTube:
<https://youtu.be/6jwLqSDRHgU>

6. VALUTAZIONI SUI RISULTATI OTTENUTI E CONCLUSIONI

6.1 Considerazioni Finali

L'applicazione è stata sviluppata affinché possa fungere come strumento di supporto per i manager di una catena di negozi di giocattoli, ma può essere altresì impiegata da altre imprese commerciali e non, che presentano un formato di data-set analogo.

Vantaggi dell'applicazione

- **Monitoraggio dei risultati nel lungo periodo:**

L'applicazione consente di controllare i risultati aziendali nel lungo periodo.

Questo monitoraggio aiuta a identificare tendenze facilitando la programmazione anticipata del rifornimento e la gestione delle scorte, migliorando così la soddisfazione della domanda dei clienti.

- **Efficienza Temporale:**

Il tempo necessario per la generazione di un report semplice può variare da pochi secondi a diverse decine di secondi. Questo è particolarmente vantaggioso se si considera che l'esecuzione manuale dello stesso compito potrebbe richiedere diversi giorni lavorativi.

L'efficienza temporale non solo libera risorse umane per altre attività strategiche, ma consente anche ai manager di reagire rapidamente a eventuali cambiamenti nel mercato.

- **Analisi del punto di pareggio:**

Questa analisi permette ai manager di farsi un'idea sui profitti attesi e sul ricavo ideale da conseguire per ogni prodotto, affinché si possa raggiungere un certo risultato operativo.

- **Analisi dell'impatto:**

L'applicazione permette di analizzare l'impatto che si avrebbe sul reddito operativo se si variassero certi fattori. Tale analisi consente ai manager di decidere se apportare modifiche strategiche.

- **Presentazione visiva dei risultati:**

I risultati sono stati principalmente presentati tramite grafici e tabelle per poter rendere le informazioni facilmente interpretabili per i decisori.

Questo approccio visivo permette di cogliere chiaramente eventuali tendenze e anomalie nei dati, facilitando così l'identificazione di aree di miglioramento e opportunità di crescita.

Criticità

La maggior criticità che si riscontra nell'applicazione risiede nei limiti dell'analisi CVP, come descritto nella sezione 2.

Questi limiti possono influenzare la precisione delle previsioni e delle decisioni strategiche, rendendo necessario l'utilizzo di altre tecniche per supportare le loro decisioni.

Conclusioni

L'applicazione rappresenta un valore aggiunto significativo per le aziende che desiderano migliorare la loro capacità decisionale e conseguire risultati soddisfacenti.

Esso consente ai manager di prendere decisioni strategiche orientati alla crescita sostenibile in un ambiente commerciale sempre più competitivo.

7. LICENZA

Questa relazione tecnica è distribuita con licenza Creative Commons BY-NC-SA 4.0

Tu sei libero di:

- **Condividere** - riprodurre, distribuire, comunicar al pubblico, esporre in pubblico, rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato.
- **Modificare** - remixare, trasformare il materiale e basarti su di esso per le tue opere

Alle seguenti condizioni:

- **Attribuzione** - devi riconoscere una menzione di paternità adeguata, fornire un link alla licenza e indicare se sono state effettuate delle modifiche. Puoi fare ciò in qualsiasi maniera ragionevole possibile, ma non modalità tali da suggerire che il licenziante avalli te o il tuo utilizzo del materiale
- **Non commerciale** – non puoi utilizzare il materiale per scopi commerciali.
- **Stessa licenza** – se remixi, trasformi il materiale o ti basi su di esso, devi distribuire i tuoi contributi con la stessa licenza del materiale originario.

Per visualizzare una copia di questa licenza, visitare:

<https://creativecommons.org/licenses/by-nc-sa/4.0/>