

Warehouse Manager

di Lamorgese Gianfranco e Cariello Cecilia

Mat. 682136 - 690765

[Link alla repository su GitHub](#)

Indice

1. Introduzione
2. Argomenti Affrontati
3. Datasets
4. Prerequisiti per il Funzionamento
5. Esecuzione

1 - Introduzione

Questo progetto nasce con lo scopo di migliorare l'efficienza e semplificare il lavoro dei dipendenti nel contesto di un magazzino di piccole-medie dimensioni.

Nello specifico, questo progetto rappresenta una parte di un sistema più grande nel quale dovrebbe essere integrato per sfruttare le funzionalità.

Allo stato attuale le funzioni come la generazione degli ordini (che appunto dipende da altri sistemi) sono state implementate sfruttando la casualità.

2 - Argomenti Affrontati

Gli argomenti affrontati sono:

1. **classificazione** tramite classificatore bayesiano;
2. **ricerca** in grafo mediante algoritmo A*;
3. predizione di valori attraverso **regressione lineare**.

3 - Datasets

Sono stati utilizzati due distinti dataset entrambi in file .csv situati nella cartella /dataset.

Nel primo dataset, usato per la classificazione, troviamo:

- **'Ordine'**: contiene ipotetici ordini ricevuti in passato dal magazzino;
- **'Categoria'**: la categorizzazione degli ordini in base alla presenza o meno di categorie contenenti a loro volta elementi fragili o potenzialmente fragili.

Il secondo dataset, invece, contiene esempi di consumi ottenuti da precedenti consegne, utili alla predizione del costo del carburante per la consegna; nello specifico troviamo:

- **'id'**: id del veicolo utilizzato;
- **'carico'**: quantità di merce trasportata;
- **'vento'**: assume il valore 0 quando il vento è assente/trascurabile, 1 quando è presente;
- **'orario'**: ora delle consegne, dato legato al traffico;
- **'L100KM'**: valore dei consumi rilevati per la consegna.

Inoltre, sono stati utilizzati dati acquisibili liberamente da OpenStreetMap

(<https://www.openstreetmap.org>). Questi dati vengono utilizzati dall'algoritmo di ricerca.

4 - Prerequisiti per il Funzionamento

Per il corretto funzionamento è necessario un computer che dispone di:

- **Python 3** (<https://www.python.org/downloads/>)
- **Anaconda** (<https://www.anaconda.com/products/individual>)

Dopo l'installazione di Python ed Anaconda sarà necessario installare alcuni pacchetti aggiuntivi:

- **Pandas, Numpy, Scikit-learn**: per la gestione di dataset e calcoli necessari per la classificazione e regressione.

Questi pacchetti sono già inclusi nella maggior parte delle versioni di Anaconda.

- **Networkx**: pacchetto per la creazione e utilizzo di strutture come grafi e network. Incluso in Anaconda, altrimenti installabile tramite: `pip install networkx`
- **OSMNX** (<https://osmnx.readthedocs.io/en/stable/index.html>): pacchetto che offre strumenti per l'acquisizione dei dati geospaziali da OpenStreetMap. Il pacchetto può essere installato tramite Anaconda con i comandi:
`conda config --prepend channels conda-forge`
`conda create -n ox --strict-channel-priority osmnx`
Nello specifico, il secondo comando è necessario per la creazione di un virtual environment.
- **Plotly** (<https://plotly.com/python/getting-started/>): pacchetto che offre funzionalità per la visualizzazione di particolari tipi di grafici. Installabile con il comando: `conda install -c plotly plotly=5.3.1`

Dopo aver installato i pacchetti, sarà necessario attivare il virtual environment creato attraverso Anaconda in precedenza da una shell di PowerShell o altro terminale equivalente. Il comando per fare ciò sarà: `conda activate ox`
Il risultato sarà una schermata del tipo:

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

Loading personal and system profiles took 598ms.
(base) PS C:\Users\glamo\Desktop\Warehouse Manager> conda activate ox
(ox) PS C:\Users\glamo\Desktop\Warehouse Manager> |

```

N.B. Una volta attivato il venv "ox" potrebbe essere necessario ripetere l'installazione di Plotly per acquisire altri componenti. In caso di problemi, una volta in ox, usare il comando:
`pip install plotly`

Inoltre, il programma necessita di **datasets** con struttura specifica ed in formato .csv, che devono essere situati nella cartella /dataset.

__pycache__	10/09/2021 18:04	File folder	
cache	10/09/2021 17:54	File folder	
dataset	10/09/2021 17:23	File folder	
classifier	10/09/2021 18:04	PY File	6 KB
main	10/09/2021 17:34	PY File	5 KB
ordine	10/09/2021 17:21	PY File	6 KB
regression	10/09/2021 18:04	PY File	1 KB
search	10/09/2021 17:21	PY File	3 KB

5 - Esecuzione

Una volta soddisfatti i prerequisiti si potrà avviare il programma usando il comando:

```
python main.py
```

Il risultato sarà una schermata del tipo:

```
----- Warehouse Manager -----  
Ordine Gestito n° 1  
  
L'ordine deve essere consegnato a: negozio elettronica  
  
Codice: 50841   Categoria: elettronica  
Codice: 76874   Categoria: informatica  
Codice: 54000   Categoria: elettronica  
Codice: 55494   Categoria: elettronica  
Codice: 115646  Categoria: videogiochi  
Il peso complessivo dell'ordine è pari a: 87 kg  
  
Indicazioni per l'imballaggio:  
L'ordine contiene oggetti fragili. Usare l'apposito materiale da imballaggio!
```

Questa versione del programma, come accennato in precedenza, ha scopo dimostrativo, quindi gli ordini e la destinazione sono generati casualmente dal sistema.

Gli ordini vengono generati dal sistema nell'ordine in cui devono essere consegnati.

L'ordine presenta dei valori utili per l'operatore, come il codice dei prodotti, la destinazione e la categoria di appartenenza del prodotto.

In particolare, attraverso un classificatore bayesiano, vengono fornite all'utente informazioni riguardanti il tipo di imballaggio necessario per il trasporto dell'ordine.

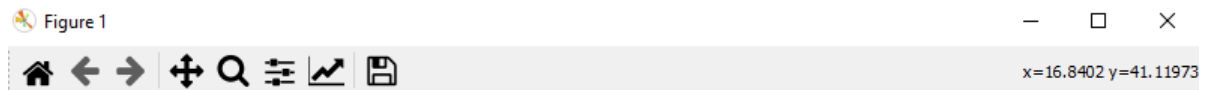
Questo eviterà all'operatore di dover controllare uno per uno i prodotti e decidere quale imballaggio usare; ovviamente, questa funzione risulta molto più utile in un contesto reale, dove gli ordini sono numerosi.

Il sistema fornirà ora all'utente una mappa. In questo caso magazzino e punti di consegna sono situati in una zona di Bari.

La mappa conterrà un percorso calcolato tramite algoritmo di ricerca A* da un punto iniziale (il magazzino o l'ultima destinazione generata, punto più a sinistra) al punto di consegna.

La mappa acquisita da OpenStreetMap viene elaborata e trasformata in un grafo: i nodi corrispondono agli incroci e sono collegati da archi (le strade), etichettati con i nomi delle strade ed accompagnati da informazioni come la lunghezza.

Da qui si potrà anche salvare una copia della mappa.



N.B. Per procedere con l'esecuzione sarà necessario chiudere la finestra contenente la mappa.

Dopo ciò il programma stampa a video informazioni sulla lunghezza del percorso e sulle strade che si dovranno percorrere per arrivare a destinazione.

```
La distanza stimata da percorrere è: 2332.2 m
Le indicazioni per andare da magazzino a negozio elettronica sono:
imbocca Viale Europa
imbocca Via Bruno Buozzi
prendi Via Francesco Crispi
gira su Via Mario Pagano
svolta su Via Brigata e Divisione Bari
Destinazione
```

Il sistema effettuerà la scelta del mezzo per le consegne e attraverso la regressione lineare, calcolerà il consumo di carburante previsto e una stima approssimativa del suo costo, basandosi sullo storico delle consegne.

```
In base alle condizioni:
Veicolo: Ford Transit
Peso: 87 kg Vento: assente/trascurabile Orario: 11

Il consumo di carburante stimato in base alle condizioni è di: 7.4 l/100km
Il costo del carburante per la consegna ammonta a: € 0.26
```

Sarà possibile decidere, in base alle informazioni ottenute, se conviene aggiungere nuovi ordini oppure fermarsi con una sola esecuzione del programma e quindi, teoricamente, effettuare la consegna.

```
Scegliere 1 per inserire un altro ordine
Scegliere 2 per terminare il programma
Scelta: |
```

Se viene scelta la prima opzione, i nuovi percorsi verranno aggiunti alla mappa e si terrà conto delle nuove informazioni per la predizione dei valori.

