

Documentazione funzionamento dei moduli

Andrea Ricci, Lorenzo Palazzetti

1 Coverage

Dato un dataset ed il numero di raccomandazioni da generare per ogni utente restituisce la coverage del modello allenato utilizzando tale dataset.

1. **Versione metodo:** chiamare il metodo `calculate_coverage` all'interno del file `coverage.py` con i seguenti parametri:

- nome del dataset
- numero di raccomandazioni da generare per ogni utente

Esempio di chiamata:

```
calculate_coverage("dataset1.csv", 10)
```

2. **Versione standalone:** eseguire da terminale il file `coverage_standalone.py`

Esempio di chiamata:

```
python coverage_standalone.py -db dataset1.csv -n 10
```

2 Diversity

Dato un dataset ed una lista di oggetti restituisce la diversity di questa lista.

1. **Versione metodo:** chiamare il metodo `calculate_diversity` all'interno del file `diversity.py` con i seguenti parametri:

- nome del dataset
- lista di oggetti

Esempio di chiamata:

```
calculate_diversity("dataset1.csv", [0000001, 00000002, 00000003])
```

2. **Versione standalone:** eseguire da terminale il file `diversity_standalone.py`

Esempio di chiamata:

```
python diversity_standalone.py -db dataset1.csv -i 0000001 0000002 0000003
```

3 Rank

Dato un dataset, un oggetto ed una lista di utenti restituisce un dataframe contenente la posizione di tale oggetto nella lista di raccomandazioni degli utenti, insieme alla media e deviazione standard dei rank.

1. **Versione metodo:** chiamare il metodo `print_rank` all'interno del file `rank.py` con i seguenti parametri:

- nome del dataset
- oggetto
- lista di utenti

Esempio di chiamata:

```
print_rank("dataset1.csv", 9999999, [0000001, 00000002, 00000003])
```

2. **Versione standalone:** eseguire da terminale il file `rank_standalone.py`

Esempio di chiamata:

```
python rank_standalone.py -db dataset1.csv -i 9999999 -u 0000001 0000002  
0000003
```

4 Consumption Rate

Dato un dataset ne restituisce due versioni modificate: una che ne mantiene tutte le colonne esistenti ed aggiunge le colonne `Sum`, `Count`, `ConsumptionRate` e `Ratio`, mentre l'altra sostituisce soltanto la colonna `Quantity` con la colonna `Modified.Quantity`, ovvero il `Ratio`.

1. **Versione metodo:** chiamare il metodo `apply_cons_rate` all'interno del file `consumption_rate.py` con i seguenti parametri:

- nome del dataset

Esempio di chiamata:

```
apply_cons_rate("dataset1.csv")
```

2. **Versione standalone:** eseguire da terminale il file `consumption_rate_standalone.py`

Esempio di chiamata:

```
python consumption_rate_standalone.py -db dataset1.csv
```

5 Print Category

Dato un dataset restituisce informazioni inerenti le categorie, in versione normale o verbose.

1. **Versione metodo:** chiamare il metodo `print_cat` all'interno del file `print_cat.py` con i seguenti parametri:

- nome del dataset
- flag per stampa della versione verbose

Di default il parametro `v` è impostato a `False`, quindi può essere omesso se non necessario.

Esempio di chiamata:

```
print_cat("dataset1.csv", v=True)
```

2. **Versione standalone:** eseguire da terminale il file `print_cat_standalone.py`

Se viene omesso il parametro `-v` viene stampata la versione "breve" dell'output e non quella verbose.

Esempio di chiamata:

```
python print_cat_standalone.py -db dataset1.csv -v
```

6 Stats

Dato un dataset restituisce informazioni generiche su di esso.

1. **Versione metodo:** chiamare il metodo `print_stats` all'interno del file `stats.py` con i seguenti parametri:

- nome del dataset

Esempio di chiamata:

```
print_stats("dataset1.csv")
```

2. **Versione standalone:** eseguire da terminale il file `stats_standalone.py`

Esempio di chiamata:

```
python stats_standalone.py -db dataset1.csv
```

7 User Stats

Dato un dataset ed una lista di utenti restituisce informazioni su di essi.

1. **Versione metodo:** chiamare il metodo `print_user_stats` all'interno del file `user_stats.py` con i seguenti parametri:

- nome del dataset
- lista di utenti

Esempio di chiamata:

```
print_user_stats("dataset1.csv", [000001, 000002, 000003])
```

2. **Versione standalone:** eseguire da terminale il file `user_stats_standalone.py`

Esempio di chiamata:

```
python user_stats_standalone.py -db dataset1.csv -u 000001 000002 000003
```

8 Item Stats

Dato un dataset ed una lista di oggetti restituisce informazioni su di essi.

1. **Versione metodo:** chiamare il metodo `print_item_stats` all'interno del file `item_stats.py` con i seguenti parametri:

- nome del dataset
- lista di oggetti

Esempio di chiamata:

```
print_item_stats("dataset1.csv", [000001, 000002, 000003])
```

2. **Versione standalone:** eseguire da terminale il file `item_stats_standalone.py`

Esempio di chiamata:

```
python item_stats_standalone.py -db dataset1.csv -u 000001 000002 000003
```

9 Graphs

Dato un dataset ed un intero n (opzionale) stampa dei grafici informativi.

1. **Versione metodo:** chiamare il metodo `print_graphs` all'interno del file `graphs.py` con i seguenti parametri:

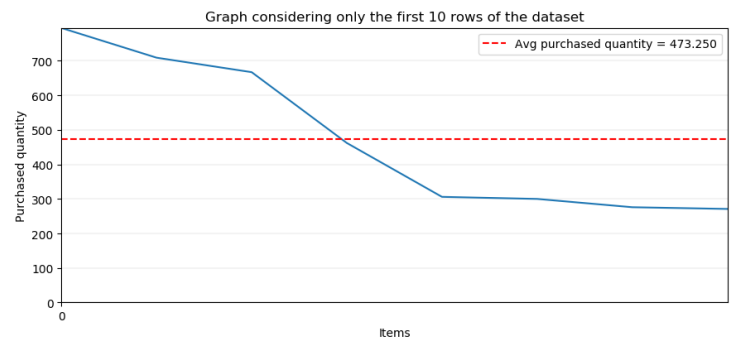
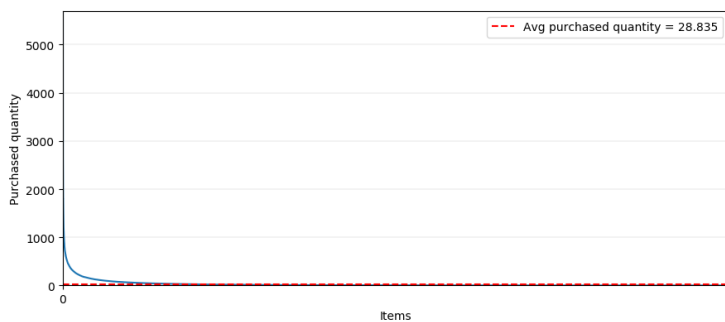
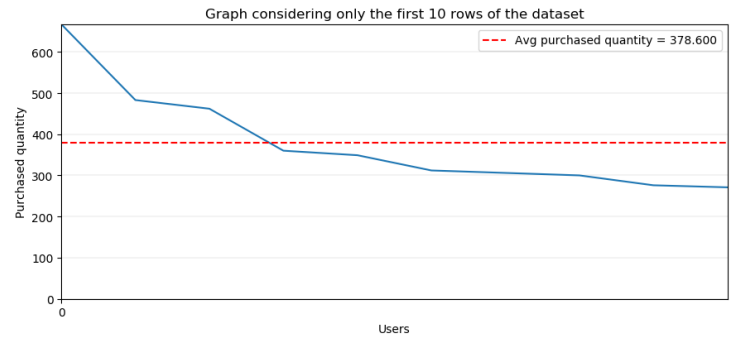
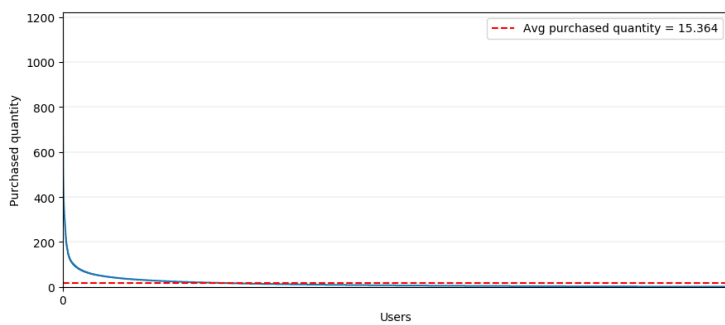
- nome del dataset
- numero di righe (opzionale)

Esempio di chiamata:

```
print_graphs("dataset1.csv", 10)
```

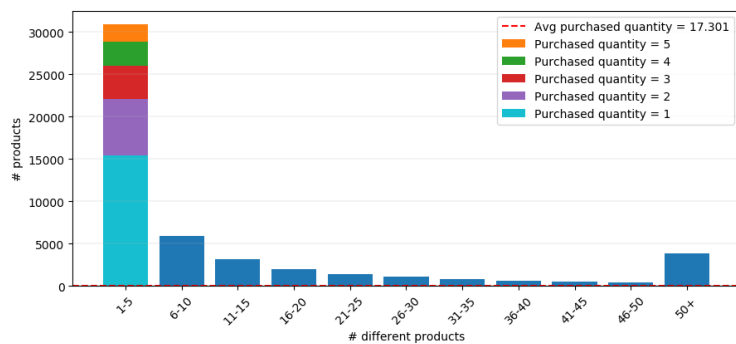
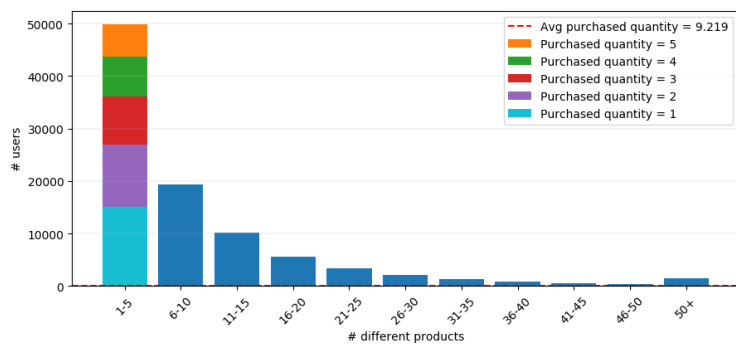
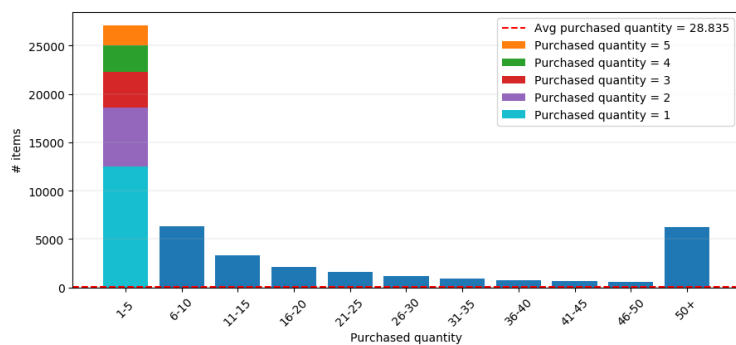
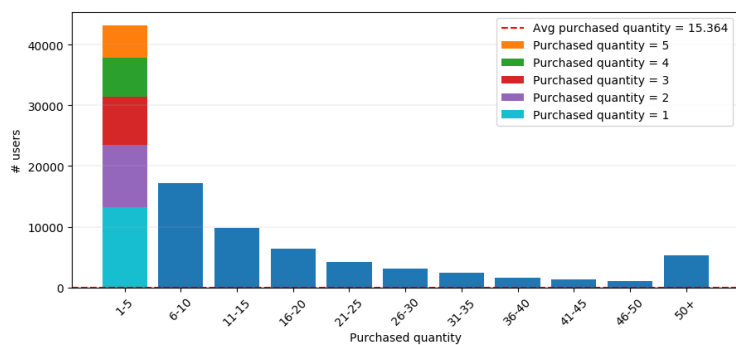
I grafici che vengono mostrati a schermo sono i seguenti:

Grafici a linea



Il numero di righe da prendere in considerazione per lo zoom, ovvero il valore n dei grafici a destra, se omesso assume il valore di default 10.

Grafici a barre



2. Versione standalone: eseguire da terminale il file graphs_standalone.py

Esempio di chiamata:

```
python graphs_standalone.py -db dataset1.csv -n 10
```