Wine Delivery

Progetto Ingegneria della Conoscenza 2020-2021,

prof. Nicola Fanizzi

Gruppo:

Laquale Alessia, 676406

Compierchio Giuseppe Pio, 678271

Introduzione

Il caso di studio si suddivide in due parti. Una parte si occupa della visualizzazione di alcuni algoritmi di ricerca per la consegna tra le varie regioni d'Italia, mostrando in base all'algoritmo selezionato (A*, BFS, DFS, ecc.) le varie differenze nella scelta del percorso. L'altra parte del progetto mantiene sempre al centro il tema del vino, ma esteso a livello mondiale, tramite una base di conoscenza, usando la regressione logistica per la classificazione multi-classe del tipo di vino (ad es. Cabernet sauvignon, merlot, malbec, ecc.) in base alle recensioni degli intenditori di vino.

WineDelivery

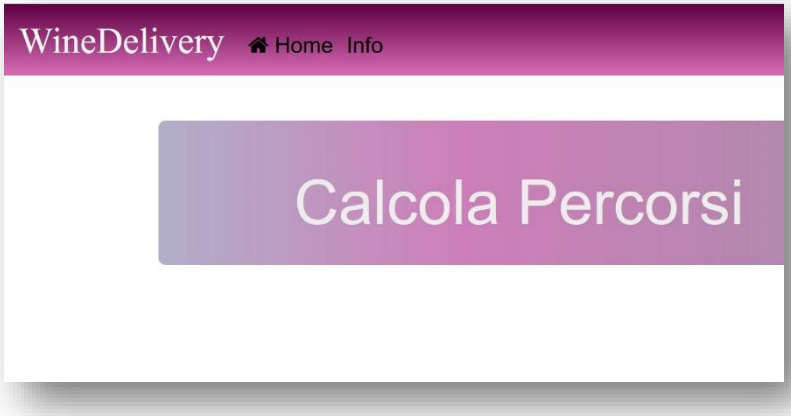
Si occupa del calcolo del percorso migliore tra i vari capoluoghi d'Italia per la consegna del vino. Sfruttando questo contesto abbiamo dato la possibilità di scegliere quale algoritmo di ricerca utilizzare ed evidenziare le diversità di scelta del percorso in base all'algoritmo scelto.

Gli algoritmi utilizzati sono stati:

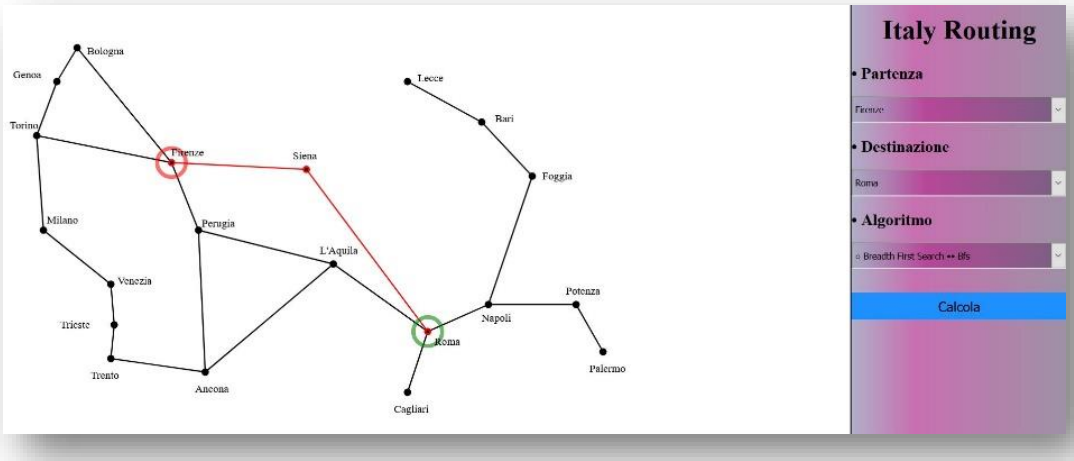
- Breadth First Search(BFS)
- Depth First Search(DFS)
- Depth Limited Search
- Iterative Depth Search
- Uniform Cost Search
- A* Search

Applicazione

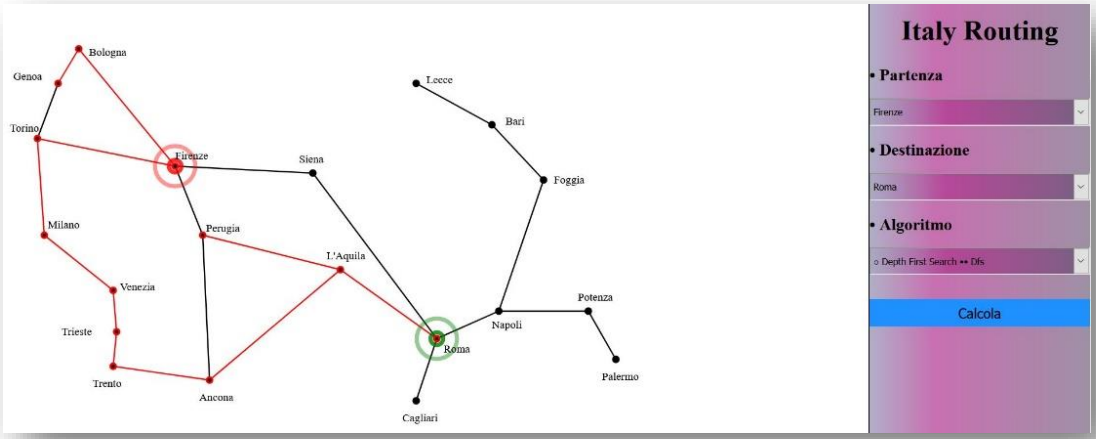
Pagina iniziale



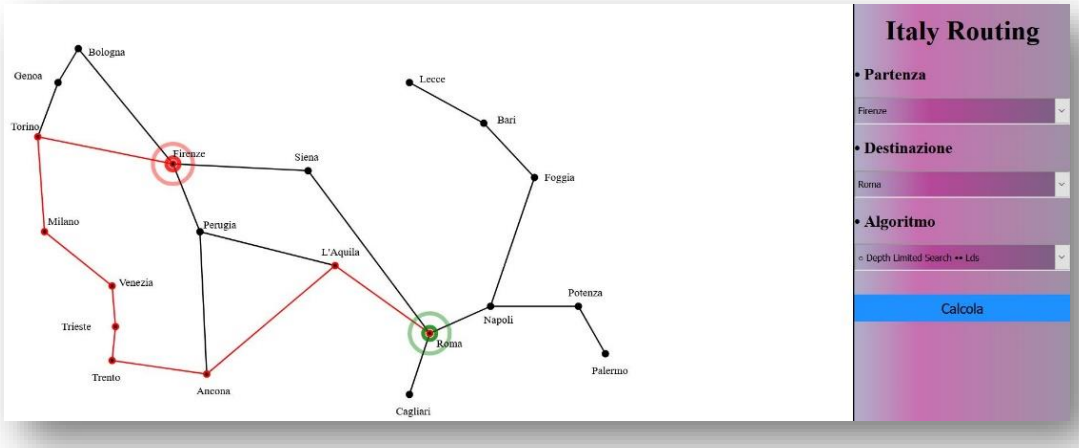
Tratta Firenze-Roma con BFS



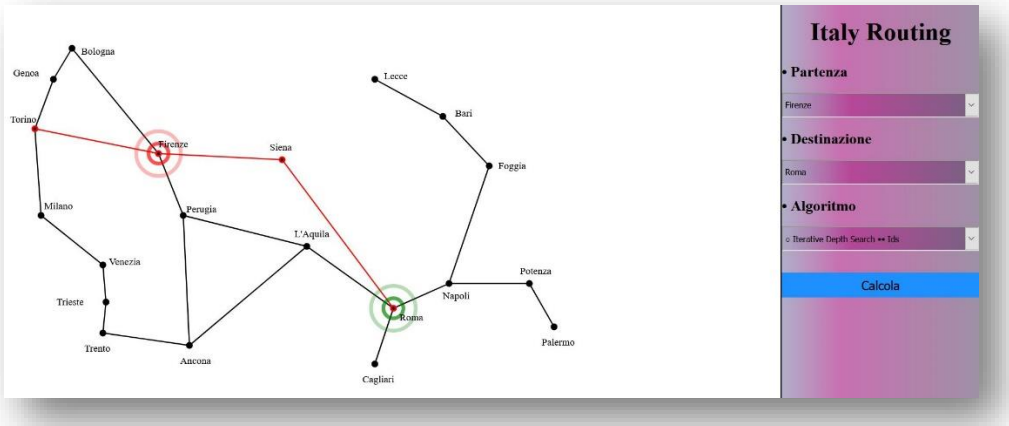
Tratta Firenze-Roma con DFS



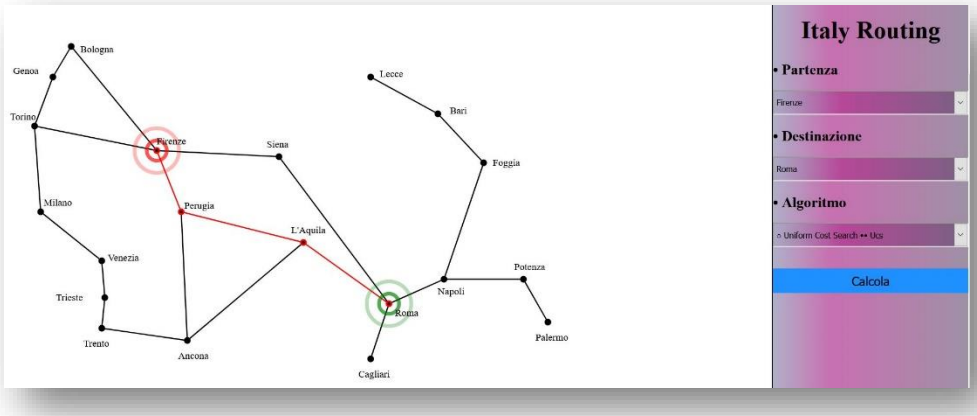
Tratta Firenze-Roma con DLS



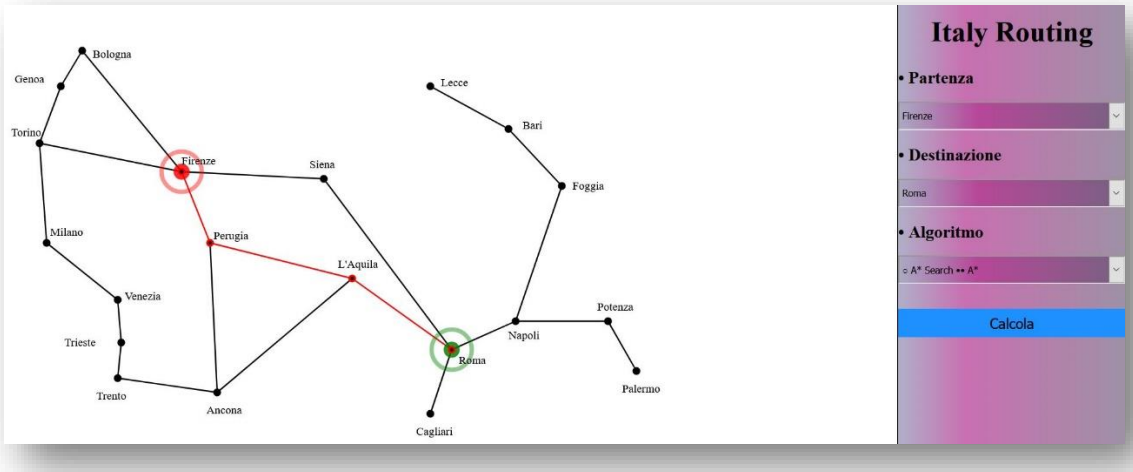
Tratta Firenze-Roma con IDS



Tratta Firenze-Roma con UCS



Tratta Firenze-Roma con A*



Algoritmi

L'idea generica alla base degli algoritmi di ricerca è di raccogliere tutti i cammini che si diramano dai nodi di partenza fino ad incontrare uno o tutti i nodi goal. L'algoritmo mantiene un insieme di cammini finora percorsi che viene detto frontiera. Inizialmente la frontiera contiene solo i nodi di partenza. Finché la frontiera non è vuota, si seleziona uno alla volta un percorso nella frontiera: se questo porta a un nodo obiettivo, restituisci il percorso; altrimenti aggiunge all'insieme tutti i percorsi ottenuti considerando gli archi uscenti dal suo ultimo nodo del percorso considerato.

BFS (ampiezza): variante della GS in cui la frontiera è una coda (FIFO). Ha complessità esponenziale nel numero di archi nel percorso della soluzione di lunghezza minima (b fattore di ramificazione, se la soluzione ha n archi, bn-1 elementi nella frontiera).

DFS (profondità): variante della GS in cui la frontiera è una pila (LIFO). Ha complessità lineare nel caso ottimo, se la soluzione si trova sul primo ramo; può divergere se ci sono rami infiniti o con cicli.

Depth-Limited Search (DLS) è un tipo di ricerca non informata. Funziona esattamente come il depth-first search, impedendo però l'inconveniente della completezza imponendo un limite alla profondità massima di ricerca. Anche se fosse possibile continuare l'espansione di un vertice a una data profondità, ciò non avverrà e di conseguenza non proseguirà andando all'infinito nella profondità di un percorso o rimanendo bloccato in un ciclo. Perciò il depth-limited search troverà una soluzione esclusivamente se è dentro un certo limite di profondità, il che garantisce quanto meno la completezza su tutti i grafi.

Iterative Depth Search (IDS) variante "limitata" della ricerca in profondità. Si compie una ricerca in profondità fino a un certo limite di archi (bound); se non si trova nessuna soluzione entro quel limite si deve rieseguire l'algoritmo. L'algoritmo parte sempre dalla frontiera dei soli nodi di partenza; richiama una sottoprocedura che compie la ricerca in profondità "limitata" passando in input il parametro che definisce la profondità massima. Questa sottoprocedura controlla se un percorso è soluzione solo quando il percorso è di profondità massima; se non lo è, continua a ricercare in profondità come DFS; se lo è, controlla se è uno stato goal: se lo è restituisce e termina; se non lo è, controlla se è stato raggiunto il limite di profondità; se è avvenuto l'algoritmo riparte; altrimenti termina (lo spazio è esaurito).

Uniform Cost Search(UCS) è un algoritmo utilizzato per spostarsi in uno spazio di ricerca ponderato diretto per passare da un nodo iniziale a uno dei nodi finali con un costo cumulativo minimo. Questa ricerca è un algoritmo di ricerca non informato poiché non prende in considerazione lo stato del nodo o dello spazio di

ricerca. Viene utilizzato per trovare il percorso con il costo cumulativo più basso in un grafico ponderato in cui i nodi vengono espansi in base al loro costo di attraversamento dal nodo radice. Questo viene implementato utilizzando una coda di priorità dove minore è il costo maggiore è la sua priorità.

A* Search unisce la misura del costo associato ad un cammino, già presente nella frontiera e considerato, ad una stima del costo associato ad un nuovo arco che si potrebbe aggiungere: f(path_n) = cost(path) + estimate(n), dove n è il nuovo nodo da aggiungere. L'algoritmo è implementato come una GS con la frontiera che è una coda con priorità ordinata in base a f. Se esiste una soluzione, A* la trova sempre e la soluzione trovata sarà ottimale se: il fattore di ramificazione è finito, i costi degli archi sono maggiori di qualche $\varepsilon > 0$ e la funzione di stima dei costi è ammissibile.

DataSet

Il Dataset preso in considerazione da repository su github contiene le varie feature:

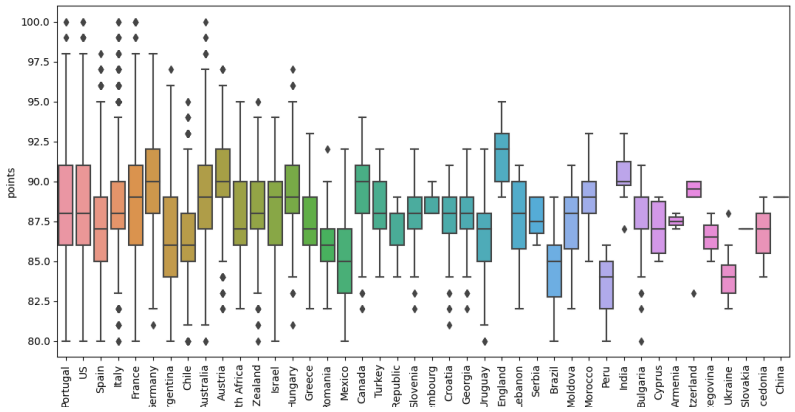
- Country: la nazione da cui proviene il vino;
- Description: la descrizione del vino, alcune frasi di un sommelier che descrivono il gusto, l'odore, l'aspetto, la sensazione al tatto del vino, ecc.
- Designation: il vigneto all'interno da cui proviene l'uva che ha prodotto il vino,
- Points: il punteggio che WineEnthusiast ha assegnato al vino da una scala da 1 a 100,
- Price: il costo per una bottiglia di vino,
- Province: la provincia o lo stato di provenienza del vino,
- Region1: l'area viticola in una provincia o stato,
- Region2: in alcuni casi viene specificata più precisamente la regione all'interno dell'area viticola ma a volta può essere vuoto,
- Tester_name: nome del tester,
- Variety: Il tipo di uva utilizzata per produrre il vino (es. Pinot Nero),
- Winery: La cantina che ha prodotto il vino.

Analisi esplorativa dei dati

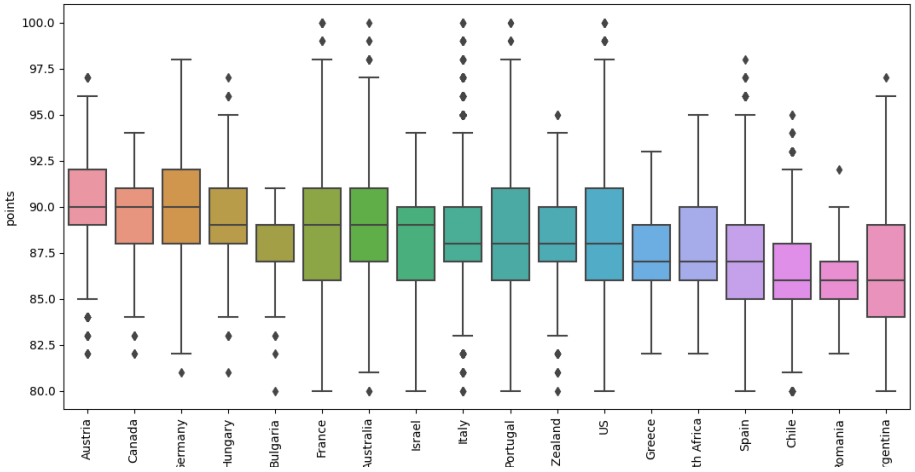
Nel Dataset cerano alcuni duplicati con le descrizioni, si è deciso di rimuoverli perché per eseguire il raggruppamento si ha bisogno solo di descrizioni univoche.

Tramite l'indice di correlazione di Pearson notiamo che esista una correlazione significativa tra il costo del vino e la sua valutazione, vale a dire che c'è un aumento medio di $ 1,18 per ogni punto di aumento della valutazione.

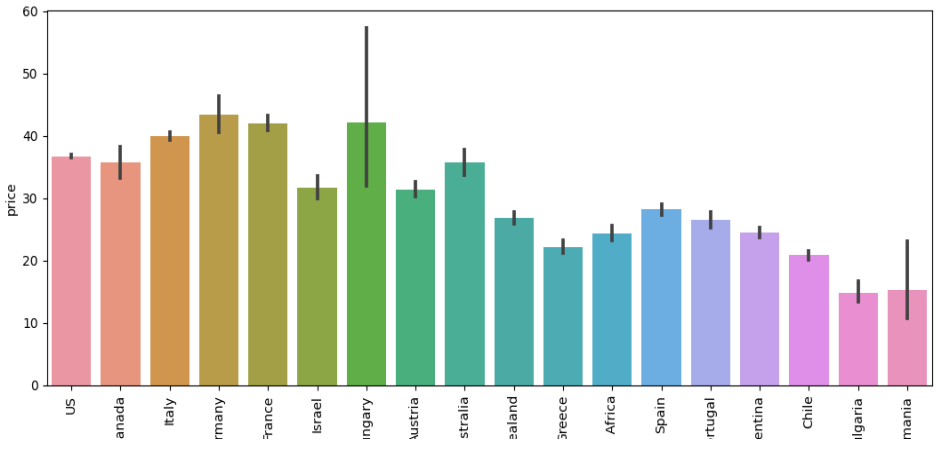
Tracciando tutti i paesi, per alcuni ci sono strani grafici a causa della bassa dimensione del campione.



Dopo aver rimosso tutti i paesi con meno di 100 osservazioni, Germania, Canada e Austria hanno i punteggi mediani più alti (points). Tuttavia, la distribuzione complessiva sembra essere abbastanza uniforme.



Di seguito sono riportati i prezzi medi del vino ordinati per mediana (dal più alto al più basso) al fine di valutare le distorsioni dei prezzi dovuti a valori anomali:

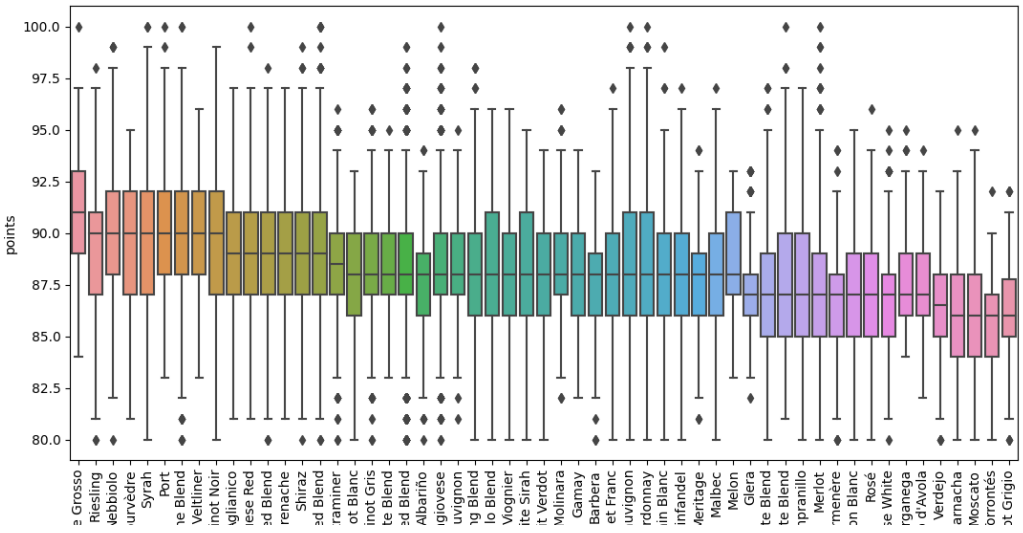


US	30.0
Canada	30.0
Italy	28.0
Germany	27.0
France	25.0
Israel	25.0
Hungary	25.0
Austria	25.0
Australia	21.0
New Zealand	20.0
Greece	19.0
South Africa	18.0
Spain	18.0
Portugal	17.0
Argentina	17.0
Chile	15.0
Bulgaria	14.0
Romania	9.0

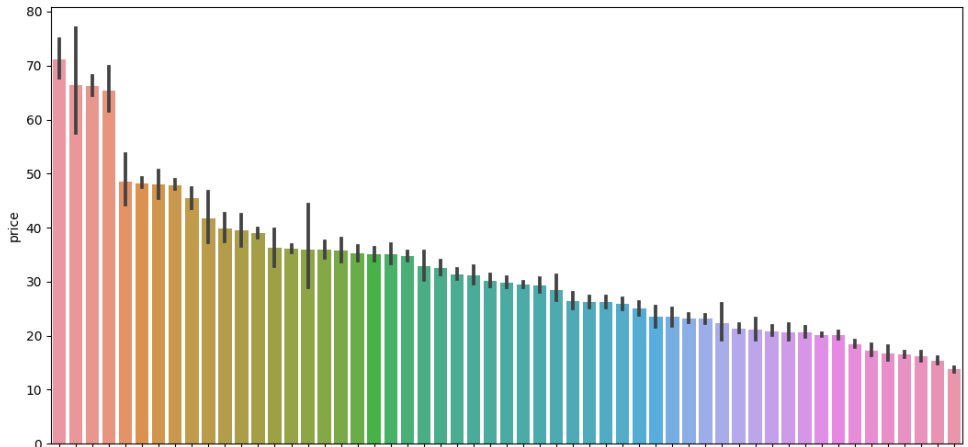
Il Dataset presenta una grande varietà di vini, ma c'è un calo esponenziale nel numero di osservazioni per ogni tipo di vino e poiché dobbiamo utilizzare queste etichette per classificare il nostro modello, abbiamo considerato solo tipi di vino con più di 200 osservazioni, perché non ci sono dati sufficienti in questi casi per generare un modello accurato per prevedere il rispettivo tipo di vino.

Di seguito è riportato un grafico a scatola contenente tutte le varietà di vino (con> 200 osservazioni) e le rispettive distribuzioni dei punti:

- Il Sangiovese Grosso sembra avere il punteggio medio più alto di tutti i vini.
- Ci sono alcuni cali interessanti che si verificano dopo il Sangiovese Grosso, Pinot Noir, Champagne Blend, Melon e Nero d'Avola.
- Di interesse è il Merlot e il Sangiovese, che tendono ad avere un gran numero di valori anomali altamente recensiti.
- Nonostante queste lievi variazioni, nel complesso la distribuzione è sostanzialmente uniforme



Per quanto riguarda il prezzo la situazione cambia, c'è una chiara variazione, che può aiutare a prevedere il tipo di vino:



Regressione Logistica

indica un modello di classificazione in cui si determinano i pesi di una funzione lineare, minimizzando un errore sugli esempi di training. I features utilizzate in questo modello sono il prezzo del vino e la sua descrizione.

Il modello restituisce una tabella con il punteggio di accuratezza: inizialmente intorno all'80% ma con l'aggiunta di alcune correzioni nelle descrizioni si è abbassato al 55%

```
Accuracy Score: 54.5360248942791 %
             actual        predicted
0             Syrah        Red Blend
1  Sparkling Blend  Sparkling Blend
2     Petite Sirah     Petite Sirah
3      Tempranillo           Malbec
4       Chardonnay  Champagne Blend
```

Linguaggi e Librerie utilizzate

- Entrambe le parti sono state implementato interamente in Python
- Le librerie utilizzate per l'implementazione del sistema sono le seguenti:
  - Pandas, matplotlib, seaborn, sklearn, Flash
- Il gruppo di lavoro ha utilizzato Git-Hub per poter lavorare in maniera più produttiva e sincronizzata.

Conclusioni

Sicuramente c'è spazio per miglioramenti e per poter includere altre funzionalità per vedere se la precisione aumenta, provando anche con classificazioni di diverso tipo. Per la WineDelivery sicuramente si potrebbe ampliare a livello mondiale fornendo una logistica più dettagliata e inserendo altre funzioni ad esempio per la scelta del veicolo.