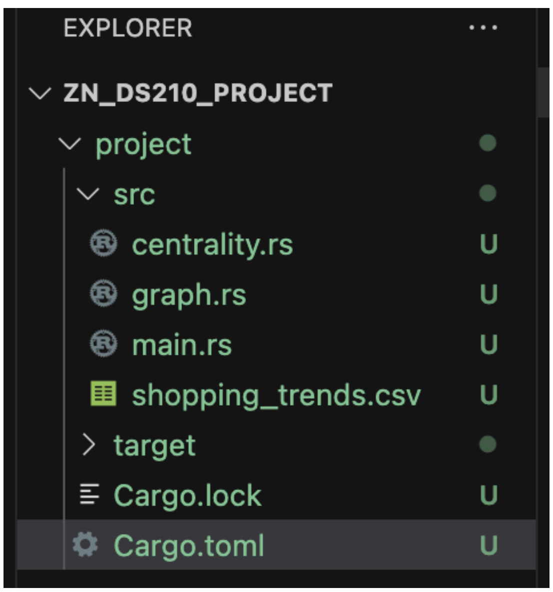
This project explores purchasing trends using a shopping trends dataset (from kaggle) that has data about customer purchases, item details, and seasonal attributes. The goal of my code/analysis is to identify the most influential products based on their connections to other items and analyze how these relationships change across different seasons.

My main research question was:
"Which products are most central (influential) in the dataset, and how does their influence vary seasonally?"

The project includes calculating degree centrality to determine the most connected items:

1. Centrality: Identifying products that are influential across the entire shopping trends dataset.
2. Seasonal Centrality: Filtering the graph by seasons (Summer, Winter, Fall, Spring) to observe how product influence changes throughout the year.

Project setup:



The src directory contains all the core files for thIS project. The main.rs file is where data is loaded from shopping_trends.csv, the graph is built, and centrality metrics (overall and seasonal) are calculated and printed. The graph.rs file handles the graph construction by defining functions to add nodes (items) and edges between nodes that share the same category. The centrality.rs file calculates centrality measures(degree centrality for the entire graph and seasonal centrality by filtering nodes based on seasons). Lastly, the shopping_trends.csv file is the main data set uploaded from kaggle. It contains the input data, such as customer purchases and item attributes which is used throughout the project.

Cargo.toml:

```
project > Cargo.toml
[package]
name = "project"
version = "0.1.0"
edition = "2021"

[dependencies]
csv = "1.1"                ✅
serde = { version = "1.0", features = ["derive"] }     ✅
ndarray = "0.16.1"    ✅
petgraph = "0.6"    ✅
```

The [dependencies] in cargo.toml lists the external libraries I had to use for the project. The csv crate (version 1.1) is used to read and compute CSV files, while serde (version 1.0 with the derive feature) supports serialization and deserialization of data structures(structs). The ndarray crate (version 0.16.1) is included for handling multi dimensional arrays. The petgraph crate (version 0.6) is used to create and manipulate graphs, which is important for this project's functionality.

Main.rs:

The main.rs file is the entry point for execution. It begins by defining the Item struct, which represents each row in the dataset with fields such as customer_id, item_purchased, category, season, and other attributes. The read_csv function reads data from the input data set shopping_trends.csv, analyzes each row, and converts it into a vector of Item structs. If a value in the CSV is invalid, it defaults to a safe value, making sure the program does not crash. In the main function, the program first calls read_csv to load the data. The graph is then built using the build_graph function from the graph module(graph.rs), which creates nodes for each unique item and edges between items sharing the same category. Once the graph is constructed, the degree centrality for each node is calculated using the calculate_degree_centrality function from the centrality module. The code then maps the node indices back to their corresponding item names and prints the centrality scores, showing how connected each item is in the overall graph. Then ,the code calculates seasonal degree centrality, which categorizes items based on their season attribute and recalculates centrality for nodes in these seasonal subgraphs. It outputs the centrality scores for each item, grouped by season, showing how influence varies across different seasons.The main.rs also includes a tests module to validate key functionalities. It uses sample data to test the accuracy of seasonal centrality, make sure duplicate items are handled correctly without creating extra nodes, and verify that no edges are created between items in unrelated categories. These tests see that the code is reliable and outputs accurate results when run.

graph.rs:

The graph.rs file is where a graph is made based on the relationships between items in the dataset. The graph is represented using the DiGraphMap type from the petgraph crate, where nodes represent unique items and edges represent relationships between items that share the same category. The create_nodes function takes a mutable reference to the graph and a list of items. It iterates through all the items and ensures that each unique item_purchased is added as a node to the graph. To avoid duplicates, it uses a HashMap to map each item name to its corresponding NodeIndex. If an item is not already in the map, a new node is added to the graph, and its index is stored in the map. The create_edges function adds edges between nodes in the graph. It iterates through all pairs of items and checks if they share the same category but are not the same item. If this condition is true, an edge is added between their respective nodes using the add_edge method. The function sees that relationships are only established between distinct items in the same category. The build_graph function combines the logic of create_nodes and create_edges to fully make the graph. It first calls create_nodes to add all the nodes to the graph, then calls create_edges to connect nodes based on shared categories. The function returns the complete graph and a mapping of item names to their corresponding NodeIndex values.

centrality.rs:

The centrality.rs file calculates centrality metrics for the graph to determine the importance or connectedness of nodes (items) within the graph. It provides two key functions: one for calculating overall degree centrality and another for calculating seasonal degree centrality. The calculate_degree_centrality function computes the overall degree centrality of nodes in the graph. It iterates through all the nodes, counts the number of neighbors (connections) for each node, and normalizes the degree by dividing it by num_nodes - 1, where num_nodes is the total number of nodes in the graph. This ensures that the centrality values range between 0 and 1, making it easier to compare the relative importance of nodes. The result is returned as a vector of floating point values, where each value corresponds to the centrality of a node. The calculate_seasonal_degree_centrality function extends this logic to analyze node centrality within seasonal subgraphs. It first categorises all unique seasons in the dataset using a HashSet. For each season it filters the list of items to include only those belonging to that season. It then maps the filtered items to their corresponding nodes in the graph using the item_node_mapping provided. After seeing the nodes for a specific season, it calculates the degree centrality for each node in the subgraph, following the same normalization process as the overall centrality. The results are stored in a HashMap, where the keys are the season

Output (cargo run):

Overall degree centrality: In this section of the output, the degree centrality of each item in the graph is calculated and displayed. Degree centrality represents how many connections a node (item) has to other nodes, normalized by the total possible connections in the graph. Items like "Blouse", "Jeans", "Sweater", and "T-shirt" have relatively high centrality scores, such as 0.4167. This indicates that these items are highly connected, meaning they share the same category with many other items. Items like "Sandals" and "Sneakers" have lower centrality scores, such as 0.1250, indicating they have fewer connections in the graph. This analysis shows which items are the most central in the dataset based on shared categories.

```
  Item 'Blouse': Degree Centrality: 0.4167
  Item 'Sweater': Degree Centrality: 0.4167
  Item 'Jeans': Degree Centrality: 0.4167
  Item 'Sandals': Degree Centrality: 0.1250
  Item 'Sneakers': Degree Centrality: 0.1250
  Item 'Shirt': Degree Centrality: 0.4167
  Item 'Shorts': Degree Centrality: 0.4167
  Item 'Coat': Degree Centrality: 0.0417
  Item 'Handbag': Degree Centrality: 0.2917
  Item 'Shoes': Degree Centrality: 0.1250
  Item 'Dress': Degree Centrality: 0.4167
  Item 'Skirt': Degree Centrality: 0.4167
  Item 'Sunglasses': Degree Centrality: 0.2917
  Item 'Pants': Degree Centrality: 0.4167
  Item 'Jacket': Degree Centrality: 0.0417
  Item 'Hoodie': Degree Centrality: 0.4167
  Item 'Jewelry': Degree Centrality: 0.2917
  Item 'T-shirt': Degree Centrality: 0.4167
  Item 'Scarf': Degree Centrality: 0.2917
  Item 'Hat': Degree Centrality: 0.2917
  Item 'Socks': Degree Centrality: 0.4167
  Item 'Backpack': Degree Centrality: 0.2917
  Item 'Belt': Degree Centrality: 0.2917
  Item 'Boots': Degree Centrality: 0.1250
  Item 'Gloves': Degree Centrality: 0.2917
```

Seasonal degree centrality: This section in the output shows degree centrality specifically for seasonal subgraphs. The graph is filtered to include only items from a specific season, and centrality scores are recalculated within that subset.

Summer: Items like Shorts, Sunglasses, T-shirt show centrality values such as 0.0105 to 0.0073. These items are relevant to the Summer season and are connected to other seasonal items in the same category.

```
Season Summer:
  Item 'Blouse': Seasonal Degree Centrality: 0.0031
  Item 'Sweater': Seasonal Degree Centrality: 0.0010
  Item 'Jeans': Seasonal Degree Centrality: 0.0105
  Item 'Sandals': Seasonal Degree Centrality: 0.0105
  Item 'Sneakers': Seasonal Degree Centrality: 0.0105
  Item 'Shirt': Seasonal Degree Centrality: 0.0105
  Item 'Shorts': Seasonal Degree Centrality: 0.0105
  Item 'Coat': Seasonal Degree Centrality: 0.0010
  Item 'Handbag': Seasonal Degree Centrality: 0.0105
  Item 'Shoes': Seasonal Degree Centrality: 0.0105
  Item 'Dress': Seasonal Degree Centrality: 0.0105
  Item 'Skirt': Seasonal Degree Centrality: 0.0105
  Item 'Sunglasses': Seasonal Degree Centrality: 0.0073
  Item 'Pants': Seasonal Degree Centrality: 0.0105
  Item 'Jacket': Seasonal Degree Centrality: 0.0105
  Item 'Hoodie': Seasonal Degree Centrality: 0.0105
  Item 'Jewelry': Seasonal Degree Centrality: 0.0010
  Item 'T-shirt': Seasonal Degree Centrality: 0.0073
  Item 'Scarf': Seasonal Degree Centrality: 0.0010
  Item 'Hat': Seasonal Degree Centrality: 0.0105
  Item 'Socks': Seasonal Degree Centrality: 0.0105
  Item 'Backpack': Seasonal Degree Centrality: 0.0073
  Item 'Belt': Seasonal Degree Centrality: 0.0073
  Item 'Boots': Seasonal Degree Centrality: 0.0105
  Item 'Gloves': Seasonal Degree Centrality: 0.0073
```

Fall: Items like Jacket, Sweater, and Boots are central in Fall, with values ranging between 0.013 and 0.0072.

```
Season Fall:
  Item 'Blouse': Seasonal Degree Centrality: 0.0103
  Item 'Sweater': Seasonal Degree Centrality: 0.0031
  Item 'Jeans': Seasonal Degree Centrality: 0.0010
  Item 'Sandals': Seasonal Degree Centrality: 0.0072
  Item 'Sneakers': Seasonal Degree Centrality: 0.0103
  Item 'Shirt': Seasonal Degree Centrality: 0.0072
  Item 'Shorts': Seasonal Degree Centrality: 0.0103
  Item 'Coat': Seasonal Degree Centrality: 0.0031
  Item 'Handbag': Seasonal Degree Centrality: 0.0072
  Item 'Shoes': Seasonal Degree Centrality: 0.0031
  Item 'Dress': Seasonal Degree Centrality: 0.0103
  Item 'Skirt': Seasonal Degree Centrality: 0.0010
  Item 'Sunglasses': Seasonal Degree Centrality: 0.0103
  Item 'Pants': Seasonal Degree Centrality: 0.0031
  Item 'Jacket': Seasonal Degree Centrality: 0.0072
  Item 'Hoodie': Seasonal Degree Centrality: 0.0072
  Item 'Jewelry': Seasonal Degree Centrality: 0.0103
  Item 'T-shirt': Seasonal Degree Centrality: 0.0010
  Item 'Scarf': Seasonal Degree Centrality: 0.0103
  Item 'Hat': Seasonal Degree Centrality: 0.0072
  Item 'Socks': Seasonal Degree Centrality: 0.0072
  Item 'Backpack': Seasonal Degree Centrality: 0.0072
  Item 'Belt': Seasonal Degree Centrality: 0.0031
  Item 'Boots': Seasonal Degree Centrality: 0.0010
  Item 'Gloves': Seasonal Degree Centrality: 0.0103
```

Winter: Central items like Coat, Scarf, Sweater, and Gloves have higher seasonal centrality, such as 0.013 or 0.0103.

```
Season Winter:
  Item 'Blouse': Seasonal Degree Centrality: 0.0103
  Item 'Sweater': Seasonal Degree Centrality: 0.0103
  Item 'Jeans': Seasonal Degree Centrality: 0.0103
  Item 'Sandals': Seasonal Degree Centrality: 0.0103
  Item 'Sneakers': Seasonal Degree Centrality: 0.0010
  Item 'Shirt': Seasonal Degree Centrality: 0.0010
  Item 'Shorts': Seasonal Degree Centrality: 0.0103
  Item 'Coat': Seasonal Degree Centrality: 0.0103
  Item 'Handbag': Seasonal Degree Centrality: 0.0103
  Item 'Shoes': Seasonal Degree Centrality: 0.0103
  Item 'Dress': Seasonal Degree Centrality: 0.0103
  Item 'Skirt': Seasonal Degree Centrality: 0.0072
  Item 'Sunglasses': Seasonal Degree Centrality: 0.0103
  Item 'Pants': Seasonal Degree Centrality: 0.0103
  Item 'Jacket': Seasonal Degree Centrality: 0.0103
  Item 'Hoodie': Seasonal Degree Centrality: 0.0072
  Item 'Jewelry': Seasonal Degree Centrality: 0.0072
  Item 'T-shirt': Seasonal Degree Centrality: 0.0103
  Item 'Scarf': Seasonal Degree Centrality: 0.0103
  Item 'Hat': Seasonal Degree Centrality: 0.0010
  Item 'Socks': Seasonal Degree Centrality: 0.0072
  Item 'Backpack': Seasonal Degree Centrality: 0.0072
  Item 'Belt': Seasonal Degree Centrality: 0.0103
  Item 'Boots': Seasonal Degree Centrality: 0.0010
  Item 'Gloves': Seasonal Degree Centrality: 0.0103
```

Spring: Items like Sneakers and Dress display centrality scores like 0.0070 or 0.0100.

```
Season Spring:
  Item 'Blouse': Seasonal Degree Centrality: 0.0100
  Item 'Sweater': Seasonal Degree Centrality: 0.0030
  Item 'Jeans': Seasonal Degree Centrality: 0.0100
  Item 'Sandals': Seasonal Degree Centrality: 0.0070
  Item 'Sneakers': Seasonal Degree Centrality: 0.0100
  Item 'Shirt': Seasonal Degree Centrality: 0.0070
  Item 'Shorts': Seasonal Degree Centrality: 0.0070
  Item 'Coat': Seasonal Degree Centrality: 0.0100
  Item 'Handbag': Seasonal Degree Centrality: 0.0070
  Item 'Shoes': Seasonal Degree Centrality: 0.0010
  Item 'Dress': Seasonal Degree Centrality: 0.0100
  Item 'Skirt': Seasonal Degree Centrality: 0.0100
  Item 'Sunglasses': Seasonal Degree Centrality: 0.0070
  Item 'Pants': Seasonal Degree Centrality: 0.0030
  Item 'Jacket': Seasonal Degree Centrality: 0.0070
  Item 'Hoodie': Seasonal Degree Centrality: 0.0100
  Item 'Jewelry': Seasonal Degree Centrality: 0.0070
  Item 'T-shirt': Seasonal Degree Centrality: 0.0100
  Item 'Scarf': Seasonal Degree Centrality: 0.0100
  Item 'Hat': Seasonal Degree Centrality: 0.0010
  Item 'Socks': Seasonal Degree Centrality: 0.0100
  Item 'Backpack': Seasonal Degree Centrality: 0.0100
  Item 'Belt': Seasonal Degree Centrality: 0.0070
  Item 'Boots': Seasonal Degree Centrality: 0.0100
  Item 'Gloves': Seasonal Degree Centrality: 0.0010
```

Observations:

1. Overall vs. Seasonal Centrality:
   Items with high overall degree centrality like T-shirts and Jeans often appear in multiple seasons with varying centrality scores. This shows their general popularity across different seasons.
2. Season-Specific Influence:
   Items like coats and scarves are central in Winter but less relevant in Summer, while sunglasses and shorts are central in the Summer.
3. Low Seasonal Scores:
   Seasonal centrality values are much smaller (like 0.0105) compared to the overall centrality because the graph in each season is smaller, containing fewer nodes and connections.

Tests (cargo test):

```
  Item "Gloves": Seasonal Degree Centrality: 0.0010
zahranensey@crc-dot1x-nat-10-239-146-192 src % cargo test
   Compiling project v0.1.0 (/Users/zahranensey/Desktop/ZN_DS210_project/project)
    Finished `test` profile [unoptimized + debuginfo] target(s) in 0.38s
     Running unittests src/main.rs (/Users/zahranensey/Desktop/ZN_DS210_project/project/target/debug/deps/project-d0184620a42f550f)

running 3 tests
test tests::test_no_edges_with_different_categories ... ok
test tests::test_duplicate_items_handling ... ok
test tests::test_seasonal_centrality ... ok

test result: ok. 3 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

zahranensey@crc-dot1x-nat-10-239-146-192 src %
                                                                              Ln 14, Col
```

1. test_no_edges_with_different_categories:
    - Purpose: makes that no edges are created between nodes (items) belonging to different categories.
    - Result: Passed, showing the graph correctly handles nodes with unrelated categories and does not create invalid edges.
2. test_duplicate_items_handling:
    - Purpose: Validates that duplicate items in the dataset do not create multiple nodes in the graph.
    - Result: Passed, showing and making sure that duplicate items are correctly handled and the graph maintains a unique node for each distinct item.
3. test_seasonal_centrality:
    - Purpose: Tests whether seasonal degree centrality calculations are accurate for items filtered by their seasons.
    - Result: Passed, also showing that the graph correctly filters nodes by season and calculates their degree centrality.

This project shows the use of graph based analysis to discover patterns in purchasing trends using a shopping dataset. By representing unique items as nodes and connecting them based on shared attributes like categories the graph shows insights into item relationships and importance. The degree centrality metric identifies globally influential items, while seasonal degree centrality highlights how product influence changes across different seasons. Through testing the code, the project makes sure that the graph construction logic is accurate, duplicate items are handled correctly, and seasonal filtering works as intended. The output shows stuff such as identifying key products that act as influencers and understanding seasonal variations.