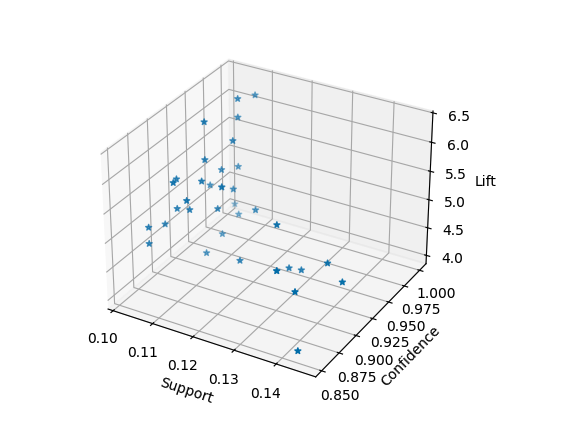
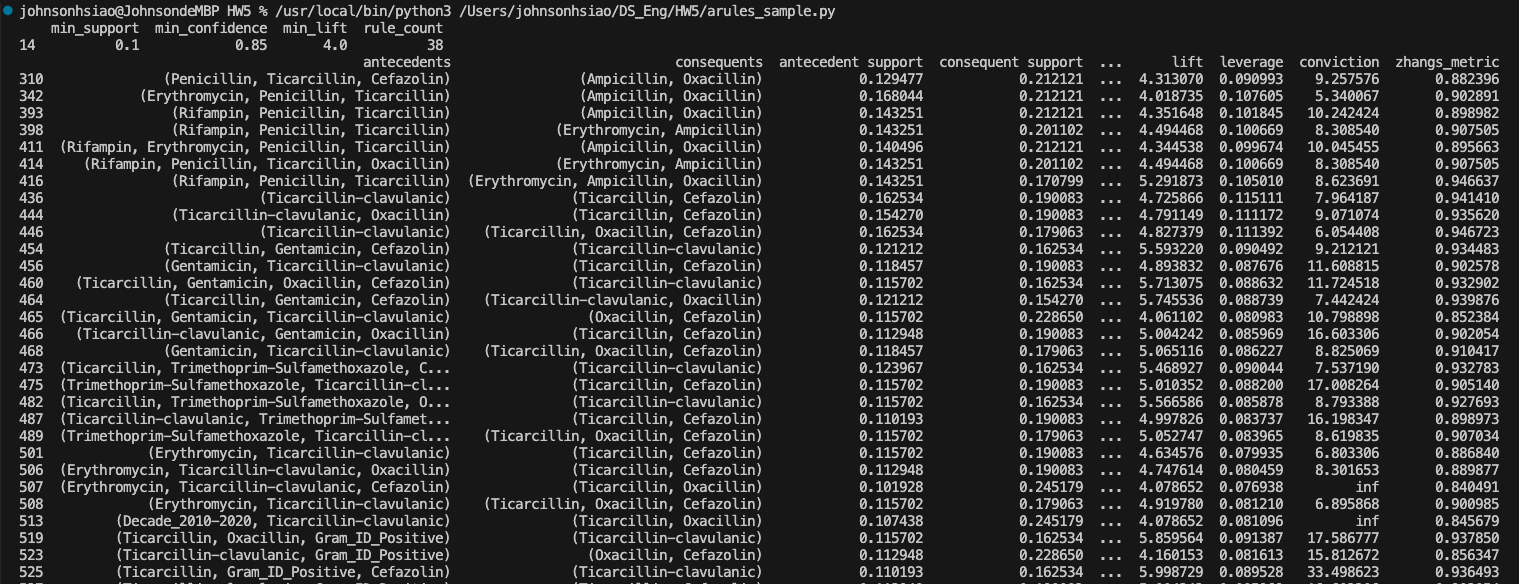
**Outputs:**

1. **filtered\_df**:
   * A DataFrame showing the combinations of min\_support, min\_confidence, and min\_lift that produce 20 to 50 rules.



1. **3D Scatter Plot**:
   * A 3D scatter plot visualizing the selected rules based on their support, confidence, and lift values. 
2. **filtered\_arules.csv**:
   * A CSV file that stores the final set of filtered high-lift association rules.
3. **Console Output**:
   * The filtered high-lift rules are also printed to the console.



**Load the Dataset (pd.read\_csv)**:

* The dataset is loaded from a CSV file (amr\_horse\_ds.csv) using pandas.

**Binning the Age Variable**:

* The Age column is binned into categories (Infant, Young, Adolescent, Adult, Senior) based on predefined age ranges using pd.cut().
* These categories make the numeric age variable more suitable for association rule mining.

**One-Hot Encoding**:

* The categorical columns (Sex, Decade, Gram\_ID, Age\_binned) are converted into binary variables using pd.get\_dummies(). This transforms the categorical data into a suitable format for mining association rules.

**Generating Association Rules**:

* Three sets of hyperparameters (min\_support, min\_confidence, min\_lift) are defined, and for each combination, frequent itemsets are generated using the FPGrowth algorithm, and association rules are extracted using association\_rules().
* The rules are filtered based on their lift values, and the number of rules that meet each set of conditions is stored in rule\_count\_summary.

**Selecting Rules Based on Count**:

* The results are stored in a DataFrame, and the rules that generate between 20 and 50 rules are selected. These parameters are used to generate the final set of high-lift rules.
* The filtered rules are then visualized in a 3D scatter plot, showing the relationship between support, confidence, and lift.