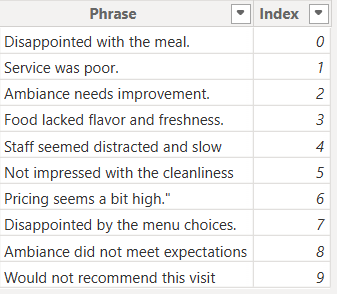
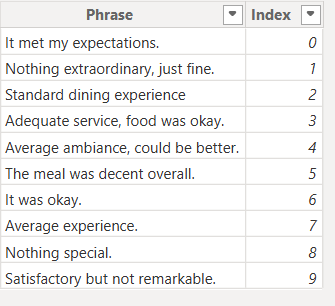
**Work Completed: Sentiment Analysis Dashboard**

**1. Phrase Tables Created:**

Three phrase tables (“Positive,” “Neutral,” “Negative”) were constructed in Power BI as separate tables. Each table lists example phrases or feedback text fragments relevant to its sentiment type.

How we created it:

* Manually entered common phrases reflecting each sentiment:
  + Positive: "Excellent service", "Loved the ambiance", etc.
  + Neutral: "It was okay", "Average experience", etc.
  + Negative: "Food was cold", "Long waiting time", etc.

Positive Phases

Neutral Phases

Negative Phases

* Imported these tables into Power BI as lookup/reference tables.

What Is the Use / Where Is It Used?

* Enriches **DynamicFeedback** Column by appending realistic feedback phrases based on detected sentiment.
* Used in DAX formulas and Power Query logic to randomly add phrases to synthetic or real feedback, providing more context or examples in visuals and test cases.

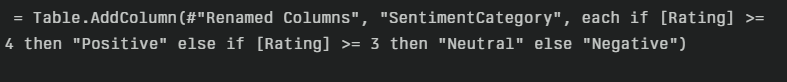
**2. Core Columns Created:**

**[A] SentimentCategory Column:**

How we Created It:

* Adds a new column called **SentimentCategory** that classifies the overall sentiment based on numerical **Rating**.
* Ratings of 4 or above are labeled **Positive**, Ratings equal to 3 are labeled **Neutral** and Ratings below 3 are labeled **Negative**.

**Example Power Query Logic:**

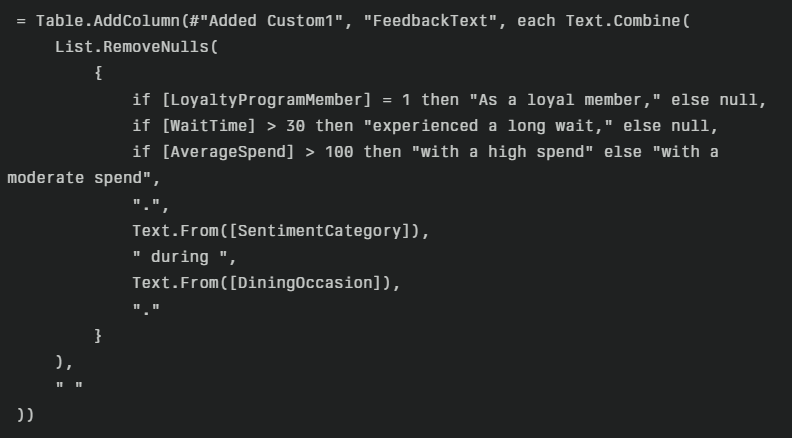


**[B] FeedbackText Column:**

How we Created It:

* Adds a new column called **FeedbackText** by combining multiple pieces of information about each customer record into a single descriptive sentence.
* For each row, it:
  + Includes phrases if the customer is a loyalty program member.
  + Mentions if the customer experienced a wait time above 30 minutes.
  + Notes if the customer's average spend is high or moderate.
  + Appends the sentiment category (Positive, Neutral, Negative).
  + Includes the dining occasion.
* Uses **List.RemoveNulls** to omit any nulls so only valid phrases are combined.
* Uses **Text.Combine** with a space separator to create a readable sentence.

**Example Power Query Logic:**

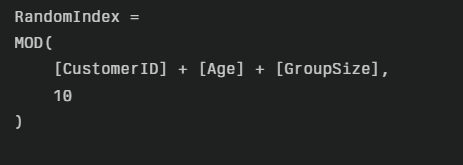


**[C] RandomIndex DAX Column:**

* This formula creates a new column named **RandomIndex** which produces a value between 0 and 9 (because of the modulus operation % 10).
* It adds together three columns: **CustomerID, Age, and GroupSize** for each row.
* The **MOD(..., 10)** function divides the sum by 10 and returns the remainder, effectively mapping the result into the range 0-9.
* This is used as a pseudo-random index—not truly random but evenly distributes rows into 10 buckets based on the combination of these fields.

**Why We Use This RandomIndex?**

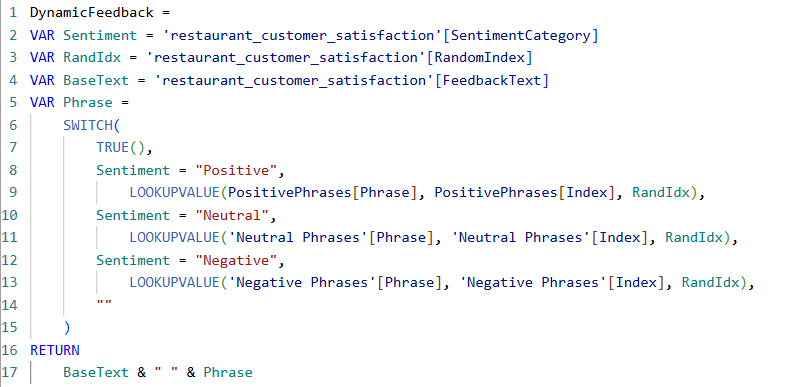
* Purpose: To select phrases from phrase tables randomly but in a reproducible way.
* Because true randomness is challenging in DAX (without volatile functions like RAND which are not ideal in calculated columns), this deterministic approach assigns a fixed "random-like" index to each row.
* This index helps in:
  + Picking a phrase from your phrase tables using this index, matching sentiment type.
  + Creating **DynamicFeedback** by appending a phrase corresponding to this index.
* Ensures variation across similar records without unpredictable changes on refresh.



**[D] DynamicFeedback DAX Column:**

How we Created It:

1. Created Supporting Phrase Tables:  
   Separate phrase tables (PositivePhrases, Neutral Phrases, Negative Phrases) were created with an indexed list of phrases related to each sentiment.
2. Used Existing Columns:
   * SentimentCategory: Classifies sentiment for each record (Positive, Neutral, Negative).
   * RandomIndex: Randomized index to pick different phrase rows.
   * FeedbackText: Base descriptive text combining customer and context info.
3. Defined Variables:
   * **Sentiment** stores the sentiment category of the current row.
   * **RandIdx** stores the random index for phrase selection.
   * **BaseText**is the original combined text field.
4. Phrase Selection (Switch + Lookup):
   * The **SWITCH** function checks the sentiment category.
   * Depending on sentiment, it uses **LOOKUPVALUE** to fetch a phrase from the relevant phrase table matching the random index.
5. Combining Strings:
   * The base feedback text is concatenated with the randomly selected phrase, creating a dynamic, enriched feedback statement.



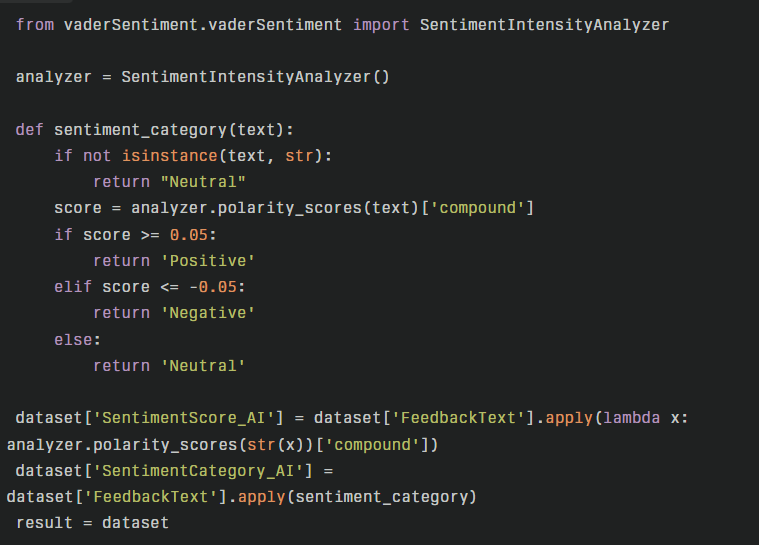
**3. AI/NLP Sentiment Classification (VADER in Python):**

**What We Did:**

* Used the VADER **SentimentIntensityAnalyzer,** a well-known NLP tool, to automatically classify and score customer feedback text for sentiment.
* Built two new columns in Power BI with the help of Power Query’s Python integration:
  + **SentimentScore\_AI:** Numeric sentiment score (from -1 negative to +1 positive).
  + **SentimentCategory\_AI:** Text label ("Positive", "Neutral", "Negative") indicating the sentiment class.

**How We Created It:**

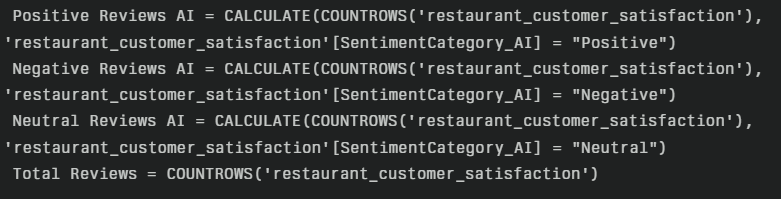
1. Configured Python Environment in Power BI:
   * Enabled Python scripting and ensured the **vaderSentiment**package was installed.
2. Loaded Data in Power Query with Python Script:
   * Selected the table containing the **FeedbackText**column as input.
3. Applied the VADER Analyzer:
   * Imported **SentimentIntensityAnalyzer** from **vaderSentiment.vaderSentiment.**
4. Defined Sentiment Scoring Logic:
   * Scored each feedback text using VADER’s **.polarity\_scores(text)['compound'].**
   * Output ranged from -1 (most negative) to +1 (most positive).
5. Created **SentimentCategory\_AI** using Python Function:
   * Defined a custom function:
     + If the text is not a string, returned "Neutral".
     + If score ≥ 0.05, labeled "Positive".
     + If score ≤ -0.05, labeled "Negative".
     + Otherwise, labeled "Neutral".
6. Populated Columns via Pandas DataFrame Operations:
   * Used **.apply(lambda x: ...)** for quick row-wise application.
   * Stored numerical scores in **SentimentScore\_AI.**
   * Stored sentiment classes in **SentimentCategory\_AI.**



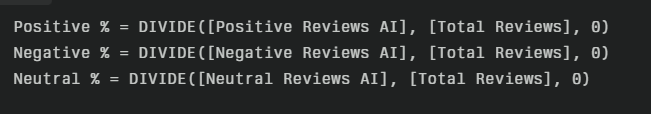
**4. DAX Measures for Sentiment Metrics**

**How We Created It:**

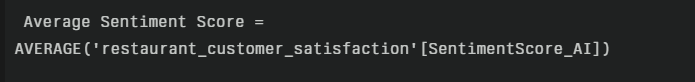
Positive/Negative/Neutral Counts:



Percentages:



Average Sentiment Score:



**What Is the Use / Where Is It Used?**

* These measures supply quantitative KPIs for cards and enable all dashboard analytics, including charts, cards, time trends, and segment breakdowns.
* Used in all visuals to compare sentiment patterns across categories, time, and filtering scenarios.

**5. Power BI Visual Built:**

**[A] Customer Sentiment Breakdown by Cuisine:**

* Type: 100% Stacked Bar Chart
* Fields Used:
  + Axis: Cuisine
  + Values: Positive Reviews AI, Neutral Reviews AI, Negative Reviews AI
* Purpose:
  + Allows direct comparison of sentiment distribution (%) for each type of cuisine.
  + Quickly reveals which cuisine receives the most positive, neutral, or negative feedback.

**[B] KPI Cards:**

* **Type:** Card Visuals
* **Fields Used:**
  + **Total Reviews**
  + **Negative %**
  + **Positive %**
  + **Neutral %**
* **Purpose:**
  + Provides headline metrics for total review count and the proportion of each sentiment type.
  + Facilitates rapid, at-a-glance overview of sentiment performance.

**[C] Average Sentiment Score Gauge:**

* **Type:** Gauge Chart
* **Fields Used:**
  + **Average Sentiment Score**
* **Purpose:**
  + Visualizes overall sentiment intensity on a scale from -1 (negative) to +1 (positive).
  + Offers a quick indicator of general customer mood.

**[D] Total Reviews by Sentiment Category:**

* **Type:** Donut Chart
* **Fields Used:**
  + **SentimentCategory\_AI**
  + **Total Reviews** (summed per sentiment)
* **Purpose:**
  + Displays the overall distribution and percentage of positive, negative, and neutral reviews.

**[E] Sentiment by Dining Occasion:**

* **Type:** 100% Stacked Column Chart
* **Fields Used:**
  + Axis: **DiningOccasion**
  + Values: **Positive Reviews AI**, **Neutral Reviews AI**, **Negative Reviews AI**
* **Purpose:**
  + Shows how sentiment varies by dining occasion (Casual, Business, Celebration).
  + Helps pinpoint which occasion prompts the best or worst feedback.

**[F] Sentiment by Visitfrequency:**

* **Type:** Stacked Column Chart
* **Fields Used:**
  + Axis: **VisitFrequency**
  + Values: **Positive Reviews AI**, **Neutral Reviews AI**, **Negative Reviews AI**
* **Purpose:**
  + Visualizes how often customers visit and the associated sentiment trends.
  + Aids in recognizing sentiment shifts based on customer visit patterns.

**[G] Slicers for Interactivity:**

* **Type:** Slicer Visuals
* **Fields Used:**
  + **MealType** (Dine-In, Takeaway)
  + **Cuisine** (American, Chinese, Indian, Italian, Mexican)
* **Purpose:**
  + Enables users to filter all visuals on the dashboard to focus on specific meal types or cuisines.
  + Makes the dashboard highly interactive and flexible for analysis.

**Business Value**

* **Comprehensive Insight:** All visuals work together to deliver a full picture of sentiment, from headline KPIs to detailed breakdowns by product, occasion, and customer behavior.
* **Interactive Analysis:** Slicers allow deep dives into segments, supporting better decision-making.
* **Clear Communication:** The use of percentage and count visuals ensures that both volume and distribution are easy to interpret.