**Sentiment Analysis on PowerBI Integrating LLM**

To carry out sentiment analysis using raw data that has not been processed in Power BI, you need to include preprocessing steps in your workflow. The overall process involves data ingestion, preprocessing, calling the sentiment analysis API, and visualizing the results. Below is a detailed guide to achieve this.

**Step-by-Step Guide**

**1. Prepare Your Environment**

1. **Set Up Power BI**:
   * Ensure you have Power BI Desktop installed.
2. **Set Up API Access**:
   * Obtain access to an LLM API that provides sentiment analysis, such as OpenAI’s GPT-3, Azure Cognitive Services Text Analytics API, or another similar service.
   * Ensure you have the necessary API keys or tokens for authentication.

**2. Ingest Raw Data into Power BI**

1. **Open Power BI Desktop**:
   * Start Power BI Desktop and create a new report.
2. **Get Data**:
   * Click on "Get Data" and choose the appropriate data source (e.g., Excel, CSV, SQL database) containing your raw customer reviews.
3. **Load Data**:
   * Load the raw data into Power BI.

**3. Preprocess Raw Data in Power Query**

1. **Open Power Query Editor**:
   * In Power BI Desktop, go to the Home tab and click on "Transform Data" to open Power Query Editor.
2. **Select and Clean Text Data**:
   * Select the column containing the raw customer reviews.
3. **Convert to Lowercase**:
   * Apply a transformation to convert all text to lowercase.
4. **Remove Special Characters**:
   * Use Power Query functions to remove special characters and punctuation.
5. **Remove Stopwords**:
   * Implement a custom function or use a predefined list to remove common stopwords.
6. **Stemming/Lemmatization** (if necessary):
   * Apply stemming or lemmatization using Power Query M language functions or by calling an external preprocessing API.

**Example Preprocessing Steps in Power Query**

m

Copy code

let

// Load the raw data

Source = YourDataSource,

// Select the column with reviews

Reviews = Source[ReviewColumn],

// Convert to lowercase

LowercaseReviews = Table.TransformColumns(Reviews, {{"ReviewColumn", Text.Lower, type text}}),

// Remove special characters

RemoveSpecialChars = Table.TransformColumns(LowercaseReviews, {{"ReviewColumn", each Text.Remove(\_, {".", ",", ";", ":", "!", "?"}), type text}}),

// Custom function to remove stopwords

RemoveStopwords = Table.TransformColumns(RemoveSpecialChars, {{"ReviewColumn", each Text.RemoveStopwords(\_, YourStopwordsList), type text}}),

// Stemming/Lemmatization if necessary

StemmedReviews = Table.TransformColumns(RemoveStopwords, {{"ReviewColumn", each Stem(\_, YourStemmingFunction), type text}})

in

StemmedReviews

**4. Call the Sentiment Analysis API**

1. **Open Advanced Editor**:
   * In the Power Query Editor, open the Advanced Editor.
2. **Write M Code to Call the Sentiment Analysis API**:
   * Here's an example M code snippet for calling a hypothetical LLM API endpoint for sentiment analysis:

m

Copy code

let

// Define the API endpoint URL

apiUrl = "https://api.example.com/v1/sentiment",

// Set up the request headers

headers = [

#"Authorization" = "Bearer YOUR\_API\_KEY",

#"Content-Type" = "application/json"

],

// Convert the preprocessed reviews to JSON

requestBody = Text.ToBinary(Json.FromValue([documents = Table.ToRecords(StemmedReviews)])),

// Make the API call

apiResponse = Web.Contents(apiUrl, [

Headers = headers,

Content = requestBody,

ManualStatusHandling = {400, 401, 403, 404, 500}

]),

// Parse the JSON response

responseJson = Json.Document(apiResponse),

documents = responseJson[documents],

// Convert the results to a table

sentimentTable = Table.FromList(documents, Splitter.SplitByNothing(), null, null, ExtraValues.Error),

expandedTable = Table.ExpandRecordColumn(sentimentTable, "Column1", {"id", "sentiment", "confidenceScores"})

in

expandedTable

**5. Transform and Load Data**

1. **Transform Data**:
   * After fetching the data from the API, you can use Power Query to transform the data as needed. For example, you might extract specific fields like sentiment scores and labels.
2. **Load Data**:
   * Load the transformed data back into Power BI for visualization.

**6. Create Visualizations in Power BI**

1. **Load the Data**:
   * Load the sentiment analysis results into Power BI.
2. **Create Visualizations**:
   * **Table Visualization**: Create a table to display the reviews along with their sentiment scores and labels.
   * **Bar Chart**: Create a bar chart to show the distribution of sentiments (positive, neutral, negative).
   * **Line Chart**: If you have timestamps for the reviews, create a line chart to show sentiment trends over time.

**Example Visualization Setup**

**Power BI Table Visualization**:

1. Go to the "Visualizations" pane and select "Table".
2. Drag and drop the fields "Review Text", "Sentiment", and "Confidence Scores" into the table.

**Power BI Bar Chart**:

1. Go to the "Visualizations" pane and select "Bar chart".
2. Drag the "Sentiment" field to the Axis.
3. Drag the "Count of Sentiment" field to the Values.

**Power BI Line Chart (if applicable)**:

1. Go to the "Visualizations" pane and select "Line chart".
2. Drag the "Date" field to the Axis.
3. Drag the "Sentiment Score" field to the Values.

**Conclusion**

By following these steps, you can integrate AI and machine learning, including large language models, to perform sentiment analysis on raw customer reviews within the Power BI platform. This approach includes preprocessing raw data to ensure it's clean and structured, making API calls to an LLM for sentiment analysis, and visualizing the results in Power BI. This comprehensive workflow enhances the analytical capabilities of your Power BI reports and provides valuable insights into customer sentiment.