Term Project 2

Data Engineering 1: SQL and Different Shapes of Data

Caroline Hamberger,

Chun-Hua Hsu,

Hanna Asipovich

Summary description:

The focus of our second term project is to check for a correlation between the consumption of mushroom pizzas and outside temperature over a one-year period. The data itself describes both pizza orders and weather during 2015 in New York, which has four distinct seasons, making it a perfect target for observation. On a technical level, we used KNIME to build our data pipeline, utilizing both MySQL and API connections. Our analysis was carried out by using linear regression predictors in KNIME. With the help of summary statistics and additional visualisation we find out that there is no statistically significant correlation between consumption of mushroom pizza and the outside temperature.

1 Overview of the Research Task

Research question: Does NYC mushroom pizza consumption in 2015 correlate to changes in temperature?

We further explored the data set used by Hanna in the Term 1 project by applying new technical processes to it. It is a fictional Kaggle[1] dataset on pizza delivered by one restaurant in the year of 2015, which includes details on daily orders, quantities and ingredients of the pizzas. In order to make use of more concepts we learned in class, we chose to complete this database with real API weather data. We explored a number of web resources before settling one found on the *Open-Meteo* website[2] which offered downloadable, free of charge, data.

The project is organized as follows. Section 2 shows the technical choices we made throughout the analysis. Section 3 describes the data models. Our analytics and visualization of this study are presented in section 4. Finally, our conclusion and discussion of this study are presented in Section 5.

2 Choice of Technical Solutions

We chose KNIME to build our data pipeline. Below our workflow is described based on our query needs:

First, with the help of **MySQL Connector** we loaded and viewed our pizza dataset in KNIME. It has already necessary data manipulation for mushroom we only run grouping by date.

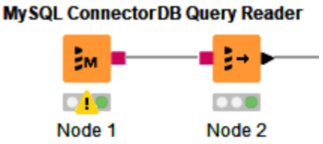
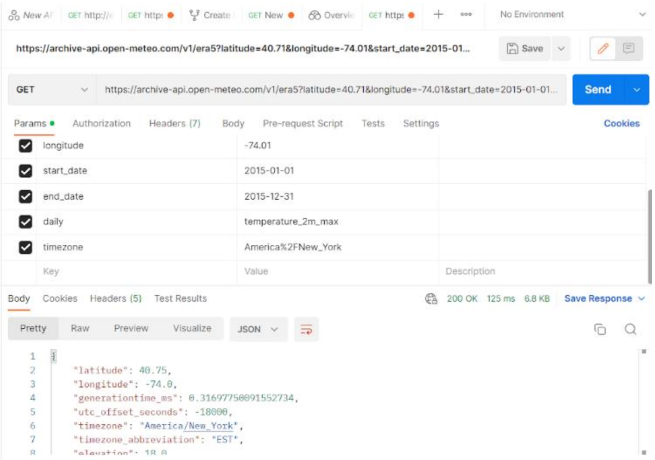


Figure 1 - Loading data from SQL

Connecting our second data set required using *Postman* to filter for only the year 2015, so that it would match our pizza database.



https://archive-api.open-meteo.com/v1/era5?latitude=40.71&longitude=-74.01&start_date=2015-01...

GET https://archive-api.open-meteo.com/v1/era5?latitude=40.71&longitude=-74.01&start_date=2015-01-01...

	Key	Value	Description
☑	longitude	-74.01	
☑	start_date	2015-01-01	
☑	end_date	2015-12-31	
☑	daily	temperature_2m_max	
☑	timezone	America%2FNew_York	

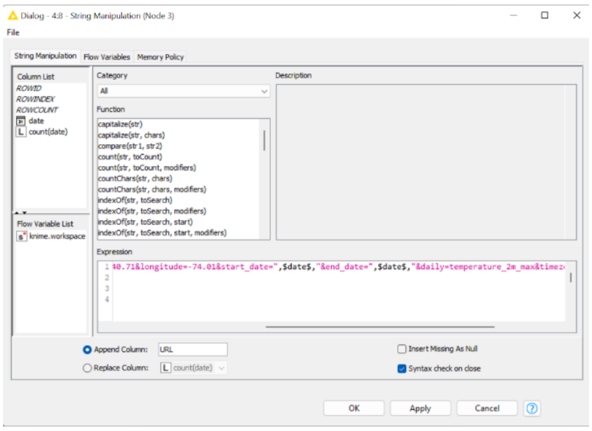
200 OK 125 ms 6.8 KB

```
{
    "latitude": 40.75,
    "longitude": -74.0,
    "generationtime_ms": 0.31697750091552734,
    "utc_offset_seconds": -18000,
    "timezone": "America/New_York",
    "timezone_abbreviation": "EST",
    "elevation": 18.0
```

Figure 2 - Using Postman for API request

[1] https://www.kaggle.com/datasets/mysarahmadbhat/pizza-place-sales?resource=download

[2] https://open-meteo.com/en/docs/historical-weather-api#latitude=42.36&longitude=-71.06&start_date=2022-10-28&end_date=2022-11-27&hourly=temperature_2m



Column List
ROWID
ROWINDEX
ROWCOUNT
date
count(date)

Flow Variable List
knime.workspace

Category: All

Function
capitalize(str)
capitalize(str, chars)
compare(str1, str2)
count(str, toCount)
count(str, toCount, modifiers)
countChars(str, chars)
countChars(str, chars, modifiers)
indexOf(str, toSearch)
indexOf(str, toSearch, modifiers)
indexOf(str, toSearch, start)
indexOf(str, toSearch, start, modifiers)

Description

Expression

```
40.71&longitude=-74.01&start_date=",$date$,"&end_date=",$date$,"&daily=temperature_2m_max&timez
```

● Append Column: URL
○ Replace Column: count(date)
□ Insert Missing As Null
☑ Syntax check on close

Figure 3 - String manipulation node to generate a string for this API request

We had to configure our API request in the KNIME on $date$:

Our next nodes in KNIME creates a GET request and takes the date column as input, sends **GET REQUEST** to the API. As the result, we get our data in **JSON format**. In the transformation part of our ETL, we will transform this data from **JSON to a table**. See graphic representation of the process in Figure 4.

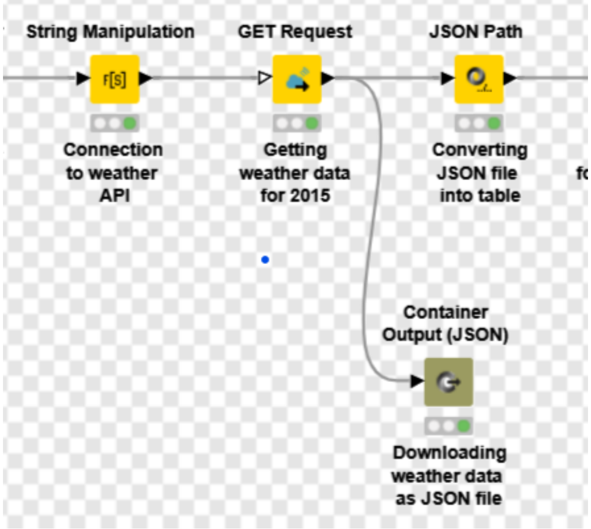
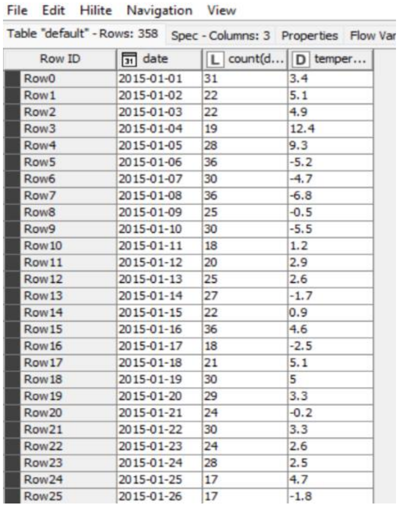

String Manipulation
GET Request
JSON Path
Connection to weather API
Getting weather data for 2015
Converting JSON file into table
Container Output (JSON)
Downloading weather data as JSON file

Figure 4 - ETL on API data



Row ID	date	count(d...	temper...
Row0	2015-01-01	31	3.4
Row1	2015-01-02	22	5.1
Row2	2015-01-03	22	4.9
Row3	2015-01-04	19	12.4
Row4	2015-01-05	28	9.3
Row5	2015-01-06	36	-5.2
Row6	2015-01-07	30	-4.7
Row7	2015-01-08	36	-6.8
Row8	2015-01-09	25	-0.5
Row9	2015-01-10	30	-5.5
Row10	2015-01-11	18	1.2
Row11	2015-01-12	20	2.9
Row12	2015-01-13	25	2.6
Row13	2015-01-14	27	-1.7
Row14	2015-01-15	22	0.9
Row15	2015-01-16	36	4.6
Row16	2015-01-17	18	-2.5
Row17	2015-01-18	21	5.1
Row18	2015-01-19	30	5
Row19	2015-01-20	29	3.3
Row20	2015-01-21	24	-0.2
Row21	2015-01-22	30	3.3
Row22	2015-01-23	24	2.6
Row23	2015-01-24	28	2.5
Row24	2015-01-25	17	4.7
Row25	2015-01-26	17	-1.8

Figure 5 - Output filtered table

After joining these data sets, our next step is to clean and tidy up the joined data set before any analysis work. We use **Column Filter** for filtering date, mushroom pizza sales and temperature, and get a very neat data table as the result. (Figure 5).

Our next phase is to create a workflow for the analytics and visuals. We use KNIME available functions and build nodes as follows: **Linear Regression Learner** + **Regression Predictor** on the dependent variable of *mushroom pizzas* and independent variable: *temperature in degrees*. We, further on, use KNIME available visualization tools to show our output results in Table view and Line Plot. For detailed analytics and visualizations see Section 4.

Below is **our complete KNIME workflow** with nodes labelled accordingly.

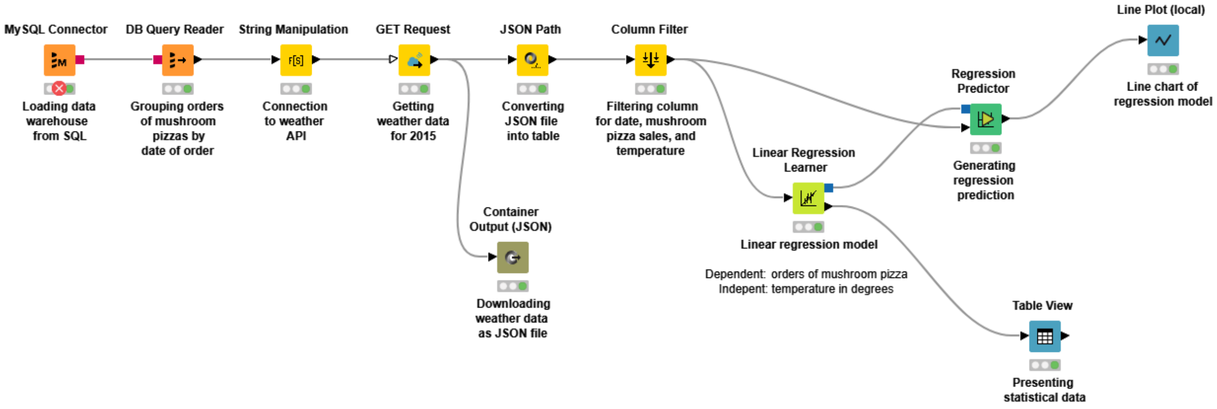

MySQL Connector
Loading data warehouse from SQL
DB Query Reader
Grouping orders of mushroom pizzas by date of order
String Manipulation
Connection to weather API
GET Request
Getting weather data for 2015
JSON Path
Converting JSON file into table
Column Filter
Filtering column for date, mushroom pizza sales, and temperature
Container Output (JSON)
Downloading weather data as JSON file
Linear Regression Learner
Linear regression model
Dependent: orders of mushroom pizza
Indepent: temperature in degrees
Regression Predictor
Generating regression prediction
Line Plot (local)
Line chart of regression model
Table View
Presenting statistical data

Figure 6 - Complete KNIME workflow

3 Data Model: EER diagram and Analytical Plan

This section shows the data model we are implementing. We use the original **pizza data set** and connect it with the API acquired **weather daily data set** for the year 2015. We connect these two on the date.

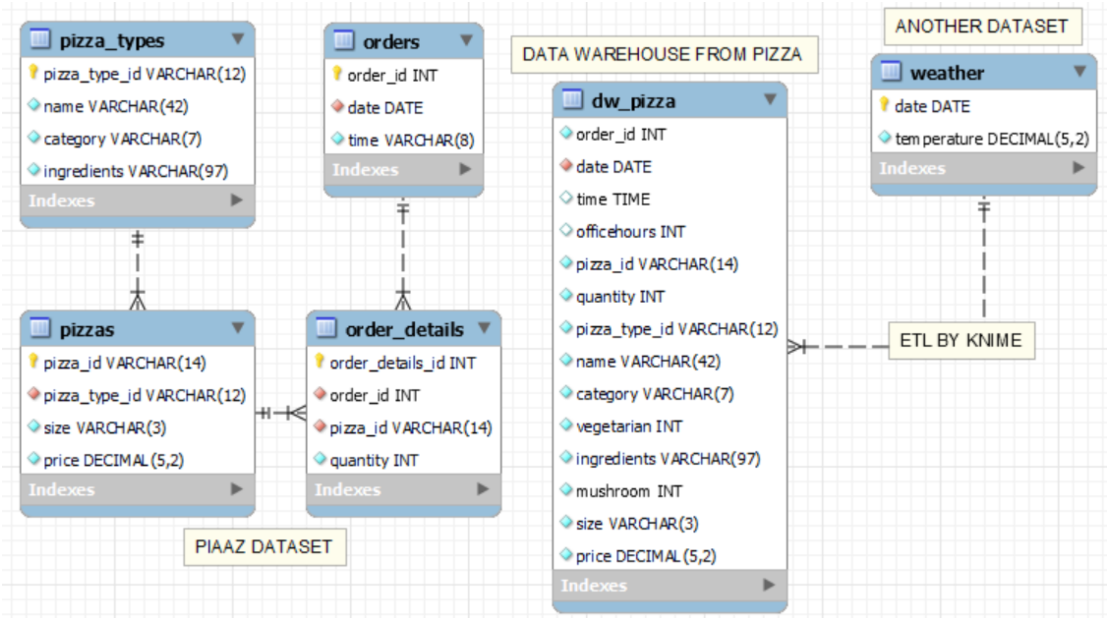


Figure 7 - EER diagram

We visualized our analytical plan as follows:

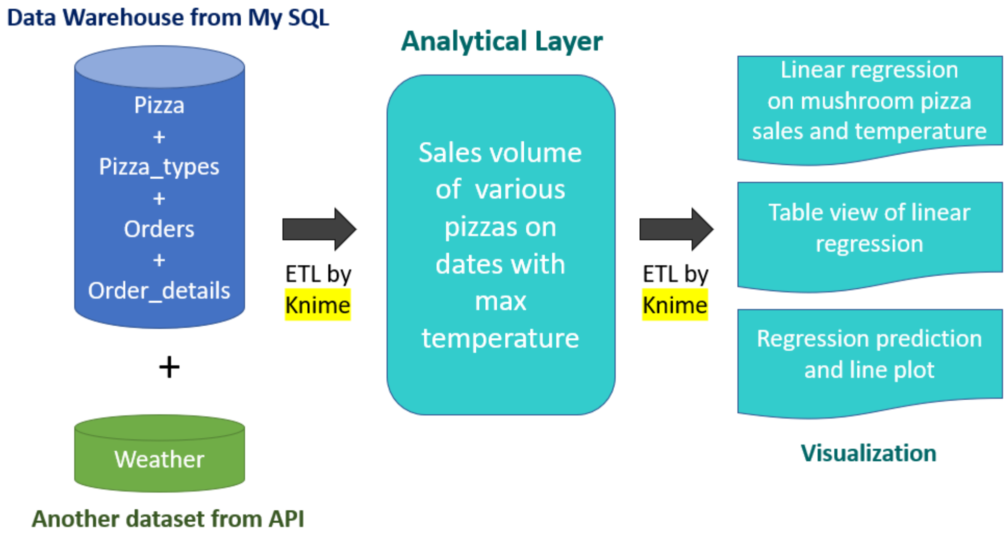

Data Warehouse from My SQL
Pizza
+
Pizza_types
+
Orders
+
Order_details
+
Weather
Another dataset from API
ETL by Knime
Analytical Layer
Sales volume of various pizzas on dates with max temperature
ETL by Knime
Linear regression on mushroom pizza sales and temperature
Table view of linear regression
Regression prediction and line plot
Visualization

Figure 8 - Analytical Plan

4 Analytics Outcome and Visualizations

We chose to use a regression model for our analytics, to best answer or question of whether the number of mushroom pizzas that were ordered depended on weather conditions. For this, we both looked at the statistical outcome, as well as the visualization for the following equation:

$$pizza\ orders = \alpha + \beta * temperature + \varepsilon$$

The outcome of the regression learner showed a slightly positive coefficient (0.019), however, at a standard error of 0.033, this result does not show and statistically significant correlation between the two values. This result is further confirmed in the visualization done via a line graph.

Figure 9 shows temperature in Fahrenheit as the green line, with a pattern following the expected seasonal trends. The red line - mushroom pizza sales - does not seem to show any similar trends. While there are a few notable spikes in mushroom pizza consumption, this cannot be correlated with temperature. The blue line simply KNIME trying to model a trend line, but as the two graphs have been shown to not be correlated, the model is entirely flat.

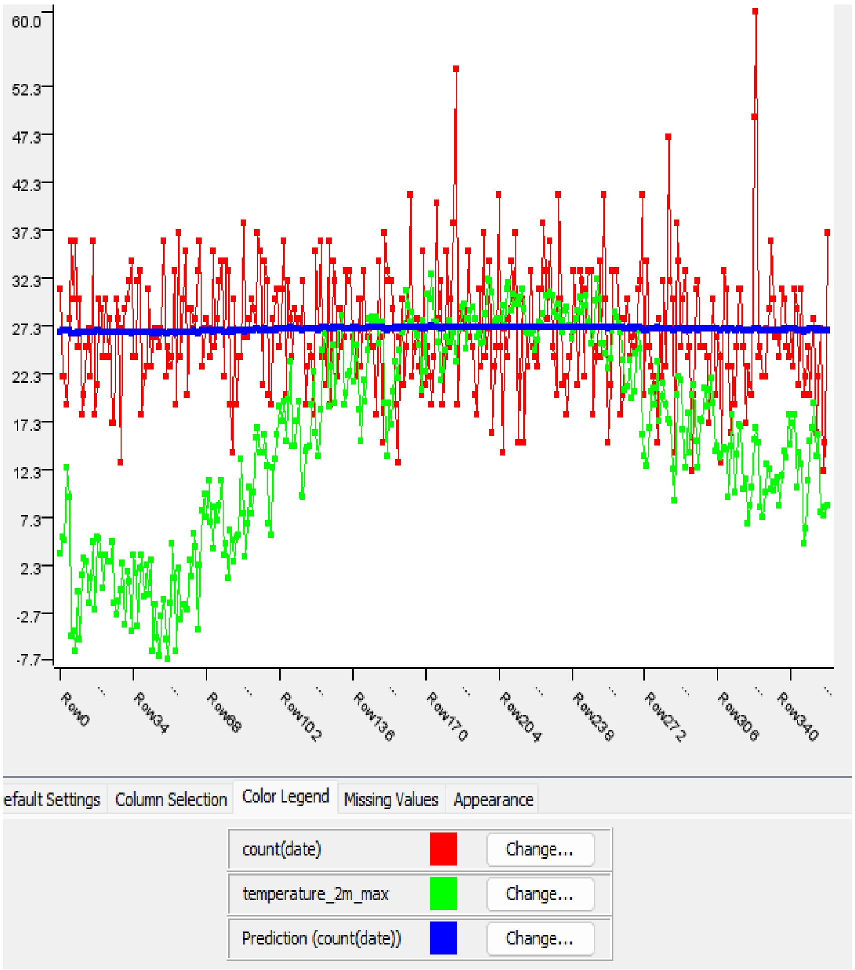


Figure 9 - Visualization of regression results

5 Conclusion

Both the summary table and line chart show that there is zero statistically significant correlation between weather data and pizza sales. This was surprising for us to see, as we initially assumed pizza orders my increase when the weather bad (i.e. colder).

KNIME showed to be a practical tool to combine our SQL data and import our API request directly. The straight-forward inherent visualization of the KNIME workflow made understanding each group member’s work process a lot easier.

6 Team Tasks completed

- Caroline Hamberger: data exploration, KNIME workflow (architecture, analytics, output), analytical report, presentation, GitHub commits.
- Chun-Hua Hsu: data exploration, KNIME workflow (data load, architecture), EER and analytical plan visualizations, API exploration, analytical report, GitHub commits.
- Hanna Asipovich: data exploration, KNIME workflow (data load), GitHub creation and commits, API configuration and Postman, analytical report.

7 Attachments on GitHub:

- Term project analytical report
- Power point presentation
- KNIME workflow file
- Data source files
- Script (or instructions) of data persistence (SQL file in case of a RDBMS). We all checked, and it worked for us!