ON THE IMPLENTATION OF A DATA PIPELINE FOR UNITED STATES NATIONAL CATTLE AUCTION DATA

by

Axel Eschholz

Report submitted in partial fulfillment of the

requirements for the Degree of

Bachelor of Computer Science

Acadia University

March, 2024

© Copyright by Axel Eschholz, 2024

The author retains copyright in this report. Any substantial copying or any other actions that exceed fair dealing or other exceptions in the Copyright Act require the permission of the author.

Table of Contents

[List of Tables v](#_Toc163085532)

[List of Figures vi](#_Toc163085533)

[CHAPTER 1: Introduction 1](#_Toc163085534)

[1.1 Background 1](#_Toc163085535)

[1.2 Motivation 2](#_Toc163085536)

[CHAPTER 2: Investigation & Design 3](#_Toc163085537)

[2.1 Requirements 3](#_Toc163085538)

[2.1.1 Data Requirements 3](#_Toc163085539)

[2.1.2 Pipeline Requirements 3](#_Toc163085540)

[2.2 Source Investigation 3](#_Toc163085541)

[2.3 Source Data 4](#_Toc163085542)

[2.4 Pipeline Design 4](#_Toc163085543)

[2.5 Tools 6](#_Toc163085544)

[2.5.1 Source Investigation - Postman 6](#_Toc163085545)

[2.5.2 Pipeline Dependencies 6](#_Toc163085546)

[2.5.3 Visualization – Tableau 7](#_Toc163085547)

[CHAPTER 3: Implementation 8](#_Toc163085548)

[3.1 Data Retrieval 8](#_Toc163085549)

[3.2 Application of Data 9](#_Toc163085550)

[3.2.1 Visualization 9](#_Toc163085551)

[3.2.2 Analysis 10](#_Toc163085552)

[3.3 Pipeline Control 10](#_Toc163085553)

[3.3.1 Extract 10](#_Toc163085554)

[3.3.2 Review 10](#_Toc163085555)

[3.3.3 Update 11](#_Toc163085556)

[3.3.4 Analyze 11](#_Toc163085557)

[CHAPTER 4: Results 12](#_Toc163085558)

[CHAPTER 5: Conclusion 14](#_Toc163085559)

[5.1 Evaluation 14](#_Toc163085560)

[5.2 Reflection 14](#_Toc163085561)

[Appendix A: Full Meta Data Report 15](#_Toc163085562)

[Appendix B: Encodings 17](#_Toc163085563)

[Glossary 19](#_Toc163085564)

[References 20](#_Toc163085565)

# List of Tables

[Table 1: Dependencies 6](#_Toc163081120)

# List of Figures

[Figure 1: USDA Data Query Dashboard 4](#_Toc163085785)

[Figure 2: Postman Interface 6](#_Toc163085786)

[Figure 3: Non-Cattle Commodities 10](#_Toc163085787)

[Figure 4: Average Price and Relative Head Count by State 12](#_Toc163085788)

[Figure 5: Price and Head Count Over Time 13](#_Toc163085789)

1. Introduction
   1. Background

In the digital age, data is a ubiquitous resource. The continued development of the Internet of Things produces an ever-growing deluge from the mass of sensors and devices connected to the internet. This unprecedented scale of information produces new sets of challenges and opportunities in the handling and analysis of what is colloquially known as ‘big data’. These data can be used to train large language models, produce highly accurate predictive analyses, and derive valuable insights at scale across just about every field and industry (Ahmed et al, 2017).

The foremost challenge in this new paradigm is that the vast majority of these data are messy and disparate, making them unusable in their raw form. Therefore, infrastructure must be developed to effectively control the flow and quality of data. This infrastructure takes the form of processes that automate the movement and control of data, known as data pipelines. Data pipelines work by first ingesting data, then processing, storing, analyzing, and performing quality control on them to produce a usable dataset for the relevant application (Cottur & Gadad, 2020). Given the varied applications of big data, the corresponding data pipelines can have wildly varying structure, complexity, and roles depending on the nature of the data and the requirements of the application. Therefore, a significant step in implementing a data pipeline is considering the data at hand and designing the automated steps to fit these requirements.

* 1. Motivation

This project is motivated by stakeholders in the finance industry concerning the implementation of a data pipeline to convert messy open-source data into insights for cattle futures markets. Futures markets are financial exchanges that facilitate the trade of standardized contracts to buy or sell commodities or financial instruments at a specified price with delivery set at a specified time in the future. The prices of these contracts, known as futures contracts, is dynamically influenced by a number of factors, including market sentiment, economic indicators, and physical market conditions of the underlying asset (Peck, 1985). However, these factors are only priced into the market to the extent that they are apparent to a majority of stakeholders. If there is a factor with the potential to influence the market, but not yet readily apparent or widely considered, it can give individual market participants an edge. The increasing digitalization of markets has amplified the importance and value of data in these decisions. A substantial quantity of data sources are openly accessible yet underutilized. This is due to their often disparate, unstructured, and raw nature, making it challenging for investors to leverage effectively.

One such case of open-source data with potential to inform futures markets is a database of cattle auction reports maintained by the United States Department of Agriculture (USDA). The cattle auction reports detail sales from all cattle auctions held across the US. The database is free to access, but, due to its structure as disparate auction house reports, it is difficult to extract meaningful data from. However, if it were to be aggregated and effectively engineered, it could potentially provide valuable insights on the future supply of cattle. Therefore, this project will attempt to design and implement a data pipeline to extract, process, and maintain a relevant dataset for the use in predicting cattle futures.

1. Investigation & Design
   1. Requirements
      1. Data Requirements

The produced data will be employed in making decisions about financial derivatives of the United States cattle market. Given that, the data must meet the following requirements.

**Completeness** – the data must encompass the vast majority of cattle auctions in the United States in order to accurately reflect macro trends in the market.

**Integrity** – the data must conform exactly to those supplied by the USDA.

**Relevance** – the dataset must reflect the most recent available data.

* + 1. Pipeline Requirements

The pipeline must be able to store, manage, and update data to reflect the requirements of the data. It must be able to do so with minimal computation power and time. Therefore, data will need to be stored in an efficient and modular architecture. The pipeline must also be able to easily and rapidly update with new reports to maintain the relevancy and recency requirement of the data.

* 1. Source Investigation

The source provided for this project by the stakeholders is a list of reports across 31 states for every cattle auction held in each state. These reports contain a weekly aggregated state summary for 27 of the 31 states. The form of each report is a PDF with tables and comments about the market conditions. Given the difficulty of extracting meaningful data from a PDF, further investigation into the source of these data was required. This investigation yielded a database lookup tool to query specific reports.

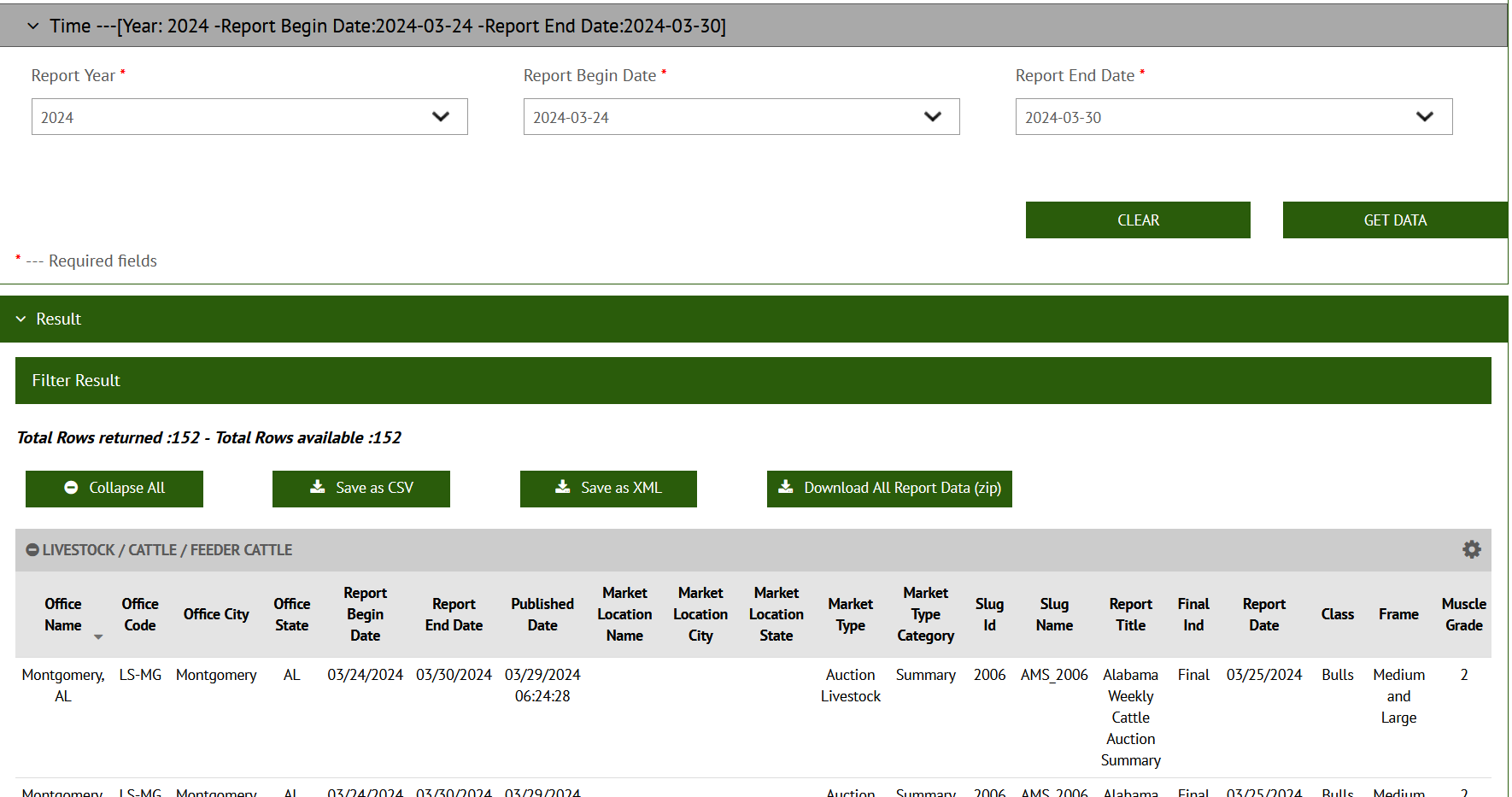


Figure 1: USDA Data Query Dashboard

By extracting the network request used by the online tool, one could retrieve report-specific data in JSON format. However, due to the limitations imposed on the browser-specific request, only 1000 rows could be returned at a time. This quota is significantly inadequate to properly retrieve data across all reports. Digging further into the source of these data reveals a separate dedicated API endpoint that can be accessed with a freely available API key that provides access to the same data but without the imposed quota.

However, this API contains all market data published by the USDA across all commodities, not just feeder and replacement cattle. In order to specify the Thus, the source for this pipeline implementation is the API combined with the report ids scraped from the original document.

* 1. Source Data

In raw form, the returned data contains 48 attributes (See Appendix A). Many of these attributes are redundant or derivative of each other (e.g. *avg\_weight\_max*, *avg\_weight\_min*). Attributes common across all instances include price, weight, and head count, with various derivatives. Other nominal attributes such as *pregnancy\_stage* are missing where not applicable. There exist ~900,000 instances across all 27 reports from January 2020 to February 2024.

* 1. Pipeline Design

The pipeline to be constructed will be based on the Extract Transform Load Transform (ETLT) model (Dayal et al, 2009). In the ETLT model the source data is retrieved and acted upon before being stored or ‘loaded’ into the data sink. They are then further processed before being displayed or used.

For the initial retrieval of the data, the python requests library will be used in conjunction with pandas to retrieve data from each report slug within a specified timeframe. Then certain attributes will be stripped or transformed to fit the shape and expectations of the data sink. Each report is stored separately to silo data and to allow for more distributed storage and processing. Therefore, the data sink consists of 27 csv files supervised by a central python interface to ensure integrity and adherence to a uniform structure. After each report is loaded into the data sink, the data is reviewed for integrity and transformed ahead of analysis and visualization.

When required for visualization and analysis, data is combined from each report silo into a single csv. These data can be filtered during this step if not all data is required for the particular analysis. The new dataset can then be applied as needed, but the data sink remains the source of truth and the new dataset can be discarded when no longer in use.

As new reports are published, the pipeline will extract the new report and merge it into the appropriate silo, ensuring that no instances are duplicated in the process. This allows for lightweight continuous updates to ensure any analysis reflects the most recent available data.

* 1. Tools
     1. A screenshot of a computer

        Description automatically generatedSource Investigation - Postman

For the source investigation, the API request tool Postman was used to analyze API endpoints and source data structure. Postman can easily store and edit request endpoints, request types, headers, authentication, and body parameters. These functionalities make it a useful tool for API analysis. Figure 2 depicts an example of the interface handling a request with returned JSON.

Figure 2: Postman Interface

* + 1. Pipeline Dependencies

The pipeline was implemented in Python due to its readability, usability, and extensive support for data analytics. The packages used in the implementation of the pipeline are enumerated below in table 1.

Table 1: Dependencies

|  |  |  |
| --- | --- | --- |
| Name | Version | Description |
| Python | 3.11.3 | Python version |
| pandas | 2.2.1 | A library for handling data in python, used extensively across the project to store all data. |
| numpy | 1.26.4 | General math and data library used in data preparation and analysis |
| requests | 2.31.0 | Used for creating API queries and retrieving initial data. |
| click | N/A | Native python library used to develop cli interface. |
| scikit-learn | 1.4.1 | Machine learning library used for the preprocessing and modelling of data. |

* + 1. Visualization – Tableau

Data visualization was conducted entirely in the Tableau visualization software. Tableau is a powerful and intuitive tool that enables easy visualization across data types.

1. Implementation
   1. Data Retrieval

Database retrieval is linked to individual report ids, referred to in the API schema as report ‘slugs’. These slugs were pulled manually from the report list and represent the 27 state summary reports being aggregated. Each slug functions as a separate endpoint that contains all historical data from all editions of that report. Query parameters were then used to specify the *report\_end\_date* attribute to retrieve data from a specific timeframe. The query was called using the python requests library, with the API key specified as the authentication parameter. The returned data was then loaded into a pandas dataframe to facilitate cleaning and storage.

url = f'https://marsapi.ams.usda.gov/services/v1.2/reports/{slug}

?q=report\_end\_date={start\_date}:{end\_date}'

response = requests.get(url, auth=(‘api\_key\_goes\_here\_xxxxxxxx’, ''))

**Code Snippet 1: API Request**

Before storing the raw data down, initial transformations were applied to fit the structure and expectations of the data sink. First, redundant and useless attributes were dropped immediately (See Appendix A). Examples of dropped attributes might include comments, which are too ambiguous to be of real use, or further iterations of market location, which are redundant to these summary data past the state code.

Next, all date-related attributes were converted to the datetime data type using the pandas.to\_datetime() method. Then all instances were sorted by *report\_end\_date* in ascending order. Once the dates were formatted and sorted, the *final\_ind* attribute was translated into a binary attribute to indicate whether the instance represents final data. All remaining nominal columns were then factorized using a uniform encoding scheme (See Appendix B). These transformations excluded the attributes *report\_title*, ­*slug\_name*, and *slug\_id* since these are administrative values used in data control but not analysis.

Once the initial transformations had been applied, the data was stored in the data sink as a csv file. Each report slug was individually retrieved, transformed, and stored. Then an integrity check was run across the entire data sink to verify that transformations had been correctly and completely applied.

* 1. Application of Data

Before conducting analysis, data from all reports was combined into a central dataset. During this process, the aforementioned administrative attributes were dropped to create a uniform set of numeric data ready for analysis and visualization.

* + 1. Visualization

Once the combined dataset was loaded into Tableau, labels needed to be linked back to the nominal data to display readable graphs. This was accomplished by converting the encodings to Tableau’s native logic and loading them as calculated values in Tableau’s data scheme. These labels allowed for insightful analysis through visualization. Figure 3 illustrates one such example where the labeled data was able to be filter to reveal a significant portion of non-cattle animals that somehow got included in the report data.

A graph with numbers and text

Description automatically generated

Figure 3: Non-Cattle Commodities

* + 1. Analysis

PCA Analysis?

Linear regression implementation?

* 1. Pipeline Control

A python-based command line interface was built to manage pipeline actions and workflows. These workflows represent the discrete common actions necessary to facilitate the pipeline operation and meet pipeline success criteria.

* + 1. Extract

In the extraction workflow the initial data is retrieved and loaded to create the data sink as detailed in section 3.1. This workflow is exceedingly time and computation heavy, so once the initial dataset is loaded it should not be required again barring any major corruption.

* + 1. Review

In the review workflow, the pipeline interface iterates through the entire data sink to ensure conformity of shape and data types. Any data type or encoding errors are corrected, where shape errors will require a re-extraction of the entire report. However, the structure of the data sink is such that offending reports can be reloaded individually without the need for a full reinitialization of the database. This workflow also summarizes the state of the database to facilitate manual maintenance and upkeep of the data.

* + 1. Update

The update workflow retrieves new data from the API and stores it down on top of the existing dataset to ensure that visualizations and analyses reflect the most recent trends. It works by checking for the last date recorded in a given report and then querying for data from that time to the present. The received data is then extracted and transformed as normal before being appended to the existing report. The update functionality eliminates the need to continually retrieve historical data and allows for the easy upkeep of the database.

* + 1. Analyze

Describes the analysis workflow.

1. Results

The data produced is a 25-attribute numerical dataset of aggregated data across 27 state summary reports. These data have been visualized using the Tableau data visualization software to derive relative insights about the US cattle market.

A map of the united states with different colored squares

Description automatically generated

Figure 4: Average Price and Relative Head Count by State

The heat map in Figure 3 illustrates the relative concentration of the US cattle market in the central states such as South Dakota, Missouri, and Oklahoma. It also displays the all-time average price of cattle across each state. By filtering for a certain timeframe, this visualization can provide a useful snapshot of the geographic distribution of the market at given time.

A graph of a graph

Description automatically generated with medium confidence

Figure 5: Price and Head Count Over Time

In a broader analysis of general market trends over time, Figure 4 graphs the total cattle sold and the average price over time. Aside from a gradual increase in price over the last four years, this graph also highlights a cyclical seasonal trend in the sale of cattle, combined with a slight falloff in the amount of cattle sold in the last year. This, combined with the recent jump in average price, might indicate supply issues.

1. Conclusion
   1. Evaluation

The design of the pipeline met success criteria of maintaining the integrity and relevance of the data in a computationally responsible manner. The functionality implemented by the pipeline facilitated the data’s relevance and integrity success criteria. The completeness criterion, however, was not achieved in full due to the exclusion of 4 states from the pipeline aggregation. While the cattle market in these states was too small to warrant a summary, the missing data still impacts the overall applicability of the dataset.

* 1. Reflection

Implementing projects for external stakeholders always involves an element of investigation and scoping to effectively meet expectations. In this case, a lack of initial investigation led to frustration in accessing source data. In the future further preliminary communication with stakeholders could mitigate such issues.

When handling field-specific data, consulting experts in the field can significantly improve the efficiency of data analysis attempts. Cross-referencing the current data with field-specific knowledge could vastly improve the quality and efficacy of analysis attempts.

# Appendix A: Full Meta Data Report

|  |  |  |  |
| --- | --- | --- | --- |
| Identifier | Description | Type | Notes |
| group | Broad category of the commodity | Nominal | e.g., Livestock |
| category | Specific category within the group | Nominal | e.g., Cattle |
| office\_name | Name of the reporting office | Nominal | Actual name of the USDA office |
| office\_code | Code identifying the reporting office | Nominal | Code for internal USDA use |
| office\_city | City where the reporting office is located | Nominal | City name |
| office\_state | State where the reporting office is located | Nominal | State abbreviation |
| report\_begin\_date | Begin date of the report period | Date | Format: MM/DD/YYYY |
| report\_end\_date | End date of the report period | Date | Format: MM/DD/YYYY |
| published\_date | Date and time when the report was published | Datetime | Format: MM/DD/YYYY HH:MM:SS |
| market\_location\_name | Name of the market location | (Missing) |  |
| market\_location\_city | City of the market location | (Missing) |  |
| market\_location\_state | State of the market location | (Missing) |  |
| commodity | Type of commodity reported | Nominal | e.g., Feeder Cattle, Slaughter Cattle |
| market\_type | Type of market | Nominal | e.g., Auction, Direct Sale |
| market\_type\_category | Category of the market type | Nominal | Further classification of market type |
| slug\_id | Unique identifier for the slug | Numeric | Integer identifier |
| slug\_name | Slug or short name for the report | Nominal | Short identifier or code for the report |
| report\_title | Title of the report | Nominal | Full title of the report |
| final\_ind | Indication if the data is preliminary or final | Nominal | "Final" or "Preliminary" status |
| report\_date | Date of the report | Date | Format: MM/DD/YYYY |
| class | Class of the cattle | Nominal | e.g., Steers, Heifers |
| frame | Frame size of cattle | Nominal | Large, Medium, Small, etc. |
| muscle\_grade | Muscle grade of cattle | Nominal | Grade based on muscle development |
| quality\_grade\_name | Name of the quality grade | Nominal | e.g., Prime, Choice |
| lot\_desc | Description of the lot | Nominal | Descriptive text of the lot |
| freight | Freight terms | Nominal | e.g., FOB, CIF |
| price\_unit | Unit of price measurement | Nominal | e.g., per head, per lb |
| age | Age of the cattle | Nominal | Specific age or age range |
| pregnancy\_stage | Stage of pregnancy for the cattle | Nominal | e.g., Early, Mid, Late |
| weight\_collect | Method of weight collection | Nominal | How the weight was collected |
| offspring\_weight\_est | Estimated weight of offspring | Numeric | Estimated weight in lbs |
| dressing | Dressing percentage | Numeric | Percentage format |
| yield\_grade | Grade based on meat yield | Numeric | Numeric grade |
| head\_count | Head count of cattle for this record | Numeric | Number of heads |
| avg\_weight\_min | Minimum average weight of cattle | Numeric | In pounds |
| avg\_weight\_max | Maximum average weight of cattle | Numeric | In pounds |
| avg\_weight | Average weight of cattle | Numeric | In pounds |
| avg\_price\_min | Minimum average price of the cattle | Numeric | Price per unit specified in price\_unit |
| avg\_price\_max | Maximum average price of the cattle | Numeric | Price per unit specified in price\_unit |
| avg\_price | Weighted average price of the cattle | Numeric | Price per unit specified in price\_unit |
| weight\_break\_low | Lowest weight in the weight break category | (Missing) |  |
| weight\_break\_high | Highest weight in the weight break category | (Missing) |  |
| receipts | Total receipts (number of heads) for auction | Numeric |  |
| receipts\_week\_ago | Receipts from the week ago period | Numeric | For comparison with current week |
| receipts\_year\_ago | Receipts from the same period a year ago | Numeric | For year-over-year comparison |
| comments\_commodity | Comments on the commodity | (Missing) |  |
| report\_narrative | Narrative summary of the report | Text | Descriptive summary of the auction data |

# Appendix B: Encodings

"office\_state": ["MS", "AL", "AR", "PA", "KY", "WV", "WY", "IN", "IL", "VA", "NM", "TN", "MO", "FL", "NE", "MT", "OK", "GA", "ND", "SD", "TX", "CO", "KS", "NC", "SC", "IA", "UT"]

"commodity": ["Feeder Cattle", "Slaughter Cattle", "Replacement Cattle", "Feeder Dairy Calves", "Slaughter Goats", "Replacement Dairy Cattle", "Feeder Pigs", "Slaughter Hogs", "Replacement Goats", "Slaughter Sheep/Lambs", "Feeder Goats", "Feeder Sheep/Lambs", "Replacement Sheep/Lambs"]

"class": ["Bulls", "Steers", "Cows", "Bred Cows", "Cow-Calf Pairs", "Bred Heifers", "Stock Cows", "Heifers", "Dairy Steers", "Dairy Heifers", "Heifer Pairs", "Wethers", "Bucks/Billies", "Kids", "Nannies/Does", "Baby Bull Calves", "Baby Heifer Calves", "Bullocks", "Pigs", "Barrows & Gilts", "Boars", "Sows", "Families", "Hair Bucks", "Hair Ewes", "Hair Breeds", "Bucks", "Ewes", "Lambs", "Wether Kids", "Wooled", "Wooled & Shorn", "Sheep", "Shorn", "Hair Lambs", "Dairy/Beef Heifers", "Dairy/Beef Steers"]

"frame": ["Medium and Large", "Small and Medium", "Large", "Medium", "Small"], "muscle\_grade": ["2", "1-2", "1", "3", "2-3", "3-4", "4"]

"quality\_grade\_name": ["Lean 85-90%", "Boner 80-85%", "Breaker 75-80%", "Prime", "Premium White 65-75%", "Number 1", "Number 3", "Utility", "Select", "Choice", "Number 2", "Choice and Prime", "Select and Choice", "Selection 2", "Utility/Non-Tubing", "Standard", "Selection 1", "Good", "Good and Choice", "Selection 1-2", "Cull", "Selection 3", "Selection 2-3", "Utility and Good", "Replacements", "Cull and Utility", "Supreme", "Approved", "Medium"]

"lot\_desc": ["Fleshy", "Fancy", "Heavy Weight", "Value Added", "Thin Fleshed", "Full", "Registered", "Replacement", "Beef Cross", "Non-Traditional", "Natural", "Pygmies", "Jersey", "Unweaned", "Light Weight", "Guaranteed Open", "Return to Feed", "Broken Mouth", "NHTC", "Roaster", "Yearlings", "Buck Lambs", "Replacements", "Buck Kids", "Non-Legible BANGS", "Gaunt", "Hair Goats", "Guernsey", "Crossbred", "Mexican Origin", "Muddy", "Poor Fleeces", "Short Docks", "Old Crop", "Source/Aged", "Spayed", "New Crop", "Canadian Origin", "Brown Swiss", "Certified Prgms", "Boar Piglets", "Ewe Lambs", "Legible Bangs", "Young", "Aged", "Ayshire", "Black Face", "Very Thin", "Dairy Goats"],

"freight": ["F.O.B."],

"price\_unit": ["Per Cwt", "Per Head", "Per Family", "Per Unit"],

"age": ["Middle Aged (5-8 yrs)", "Young (2-4 yrs)", "Young/Middle Aged (2-8 yrs)", "Aged (>8 yrs)", "(<2 yrs)", "Middle Aged/Aged (>5 yrs)", "Yearlings (1-2 yrs)", "Middle Aged (4-6 yrs)", "Kids (<1 yr)", "Aged (>6yrs)", "Lambs (<1 yr)"],

"pregnancy\_stage": ["1st Stage (1-3 mo)", "Open", "3rd Stage (7-9 mo)", "2nd Stage (4-6 mo)", "2nd/3rd Stage (4-9 mo)", "1st/2nd Stage (1-6 mo)", "All Stages (1-9 mo)", "Bred", "Spring", "Exposed", "Winter", "Summer"]

"weight\_collect": ["Actual", "N/C", "Estimate"]

"offspring\_weight\_est": ["<150", "150-300", ">300", "<20 lbs", "20-40 lbs", "40-60 lbs"]

"dressing": ["High", "Average", "Low", "Very Low"], "yield\_grade": ["1-2", "1", "3", "2-3", "2", "3-4", "1-3", "2-4", 1.0, "1.0", "U.S. 1", "4-5", "U.S. 3", "U.S. 2", "U.S. 1-2", "5", "U.S. 2-3", "3-5"]}

# Glossary

USDA – The United States Department of Agriculture

API – Application Programming Interface

# References

Ahmed, E., Yaqoob, I., Hashem, I. A. T., Khan, I., Ahmed, A. I. A., Imran, M., & Vasilakos, A. V. (2017). The role of big data analytics in Internet of Things. *Computer Networks*, *129*, 459-471.

Cottur, K., & Gadad, V. (2020). Design and development of data pipelines. *Int Res J Eng Technol (IRJET)*, *7*, 2715-2718.

Dayal, U., Castellanos, M., Simitsis, A., & Wilkinson, K. (2009, March). Data integration flows for business intelligence. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology* (pp. 1-11).

Peck, A. E. (1985). The economic role of traditional commodity futures markets. *Futures markets: their economic role*, 1-81.