ETL Report

Daniel Brickman, Sawyer Tucker, Greg Wagner, Christopher Nash
02/07/2022

Introduction

Our goal in this project is to determine how best to predict the price of common but important commodities like wheat, corn, soybeans, and gold. We also want to be able to determine the greatest producers and consumers of these goods, both domestically and internationally. To that end, a lot of data will come from disparate sources and need to be cleaned transformed into something useable for our purposes. We are taking data from an API called Commodities API that has historical data for our chosen commodities, data from the United States Department of Agriculture that will allow us to determine domestic production of our commodities, and a Foreign Agricultural Service Data API that will allow us to access data regarding import and export information for the purpose of determining leaders in international production and consumption. This data needs to be transformed because not only are there many sources for this data, but those sources also include a lot of information that does not help us make our predictions, and we would otherwise need to store useless data.

Data Sources

Economic Research Service API

United States Department of Agriculture. (2021, December 16). Economic Research Service - ARMS Data API. Retrieved February 7, 2022, from https://www.ers.usda.gov/developer/data-apis/arms-data-api/

Foreign Agricultural Service Data API

United States Department of Agriculture. (n.d.). *Foreign Agricultural Service Data APIs*. OpendataWeb. Retrieved February 7, 2022, from https://apps.fas.usda.gov/opendataweb/home

Commodities API:

Zyla Labs. (n.d.). *API: Free Oil, coffee, Wheat & commodities rates JSON API*. Commodities. Retrieved February 7, 2022, from https://www.commodities-api.com/dashboard

Extraction

Where did you get the data from? How did you get the data? What format is the extracted data? What steps were taken to extract the data? Be sure to number steps when the order matters.

Commodities API

1. Go to https://www.commodities-api.com/dashboard and follow the instructions to set up your API key.
2. A key will be sent to your email. Save this key in a variable and encrypt it
3. Import json, pandas as pd, certifi, and urllib3. Make sure spark is working on your Databricks
4. Create http variable that will be used in step .
   ```
   http=urllib3.PoolManager(cert_reqs='CERT_REQUIRED',ca_certs=certifi.where())
   ```
5. Create a variable that stores the URL for the most updated prices in the api (latest) and the symbols needed to get the data you want.
6. Create a variable using pandas that gets the API data. `step5=http.request('GET', step4variable)`. This will get a JSON dataset stored in your variable.
7. Check status of previous step, should return 200
8. Create a variable that loads the JSON dataset from step 3 into a dictionary.

USDA Agricultural Resource Management Survey (ARMS)

1. Go to https://www.ers.usda.gov/developer/data-apis/arms-data-api/ and follow the instructions to set up your API key.
2. A key will be sent to your email. Save this key in a variable and encrypt it.
3. Import requests and pandas, and make sure spark is working on your Databricks
4. Using requests.get(), get the proper URL with filtered variables in the following steps:
   a. The base URL is https://api.ers.usda.gov/data/arms/surveydata?api_key={APIKEY}
   b. Using the & symbol, add the following parameters:
      i. Year=2020,2019,2018,2017,2016,2015,2014,2013,2012,2011
      ii. State=mn
      iii. Category = sal
      iv. Category2=spec
      v. report=structural+characteristics
      vi. Variable=kount
5. Save the json version of the response to a variable, then create a pandas dataframe based on it
6. Set up a second get query with the year only for 2020, looking at all states and finding the total value of output (vprodtot) by spec (the report will stay the same) Set spec as the first category, as there will be no category2.
7. Save as a json variable and then turn it into a pandas dataframe
8. Filter the dataframe down to just state, specialty, and estimate and rename columns accordingly.
9. Turn into spark dataframe and save to the azure blob in a new directory of the same gold-standard container.

USDA Foreign Agriculture Export Data

1. Go to https://apps.fas.usda.gov/opendataweb/home and sign up for an API key
2. The countries codes will be looking at are US, IN, BR, and CH. Save these strings in a list to iterate, and add more countries if needed
3. Set up a blank list to where the data will be saved in in dictionary form
4. Use requests.get to look up the H6SCodes of all commodities from the GATS API, and save to a variable in json form
5. Turn this list into a pandas dataframe

6. We need all the codes involving wheat, corn, and soybean in their own lists for later. In the current dataframe, filter the rows down to each commodity type respectively by filtering by the name. Save the codes into each respective list.
7. Merge each commodity list together into one variable called coms_list
8. Set up a nested for loop: The first loop iterates over the range of years between 1994 and 2017. Immediately after writing that line, write another for loop going over each countrt code in the country list
9. Use the /api/gats/UNTradeExports/reporterCode/ function, while inputing the country code and year where needed, to create a row of data in the loop. Combine the year, country code, commodity code, and value into a dictionary and append it to the main data list that is currently blank
10. Save the final output as a spark dataframe once the loops are complete (this will take a few minutes!)

Transformation

Commodities API

1. Transform dictionary created in the last section to a pandas data frame.
2. Drop unnecessary columns (success, base, unit, reates.USD, date)
3. Rename columns. For this project we renamed rates.CORN to corn, rates.WHEAT to wheat, rates.SOYBEAN to soybean, and rates.XAU to gold.
4. Create an additional timestamp column which converts the Unix time format to a Y-M-D h:m:s format. We also converted the time from UTC to CST to better match the geographical area for the members of this group.
5. Multiply the corn data by 100.
6. Convert the pandas data frame to a spark data frame.

USDA Agricultural Resource Management Survey (ARMS)

1. Filter the dataframe to only include the columns ['year', 'category_value', 'category2_value', 'estimate']
2. Rename the category/category2 columns "Income Class" and "Specialty" respectively
3. Create Spark dataframe from pandas dataframe
4. Repeat this process for the output value dataframe, naming the columns accordingly.

USDA Foreign Agriculture Export Data

1. Group the dataframe by year, country, and commodity using groupby(), the use sum() to sum the value column
2. Filter the commodity column to only include the wheat codes by using isin(wheat_codes), then change those values to equal 'Wheat'
3. Repeat step 2 for 'Corn' and 'Soy'

Load

If you were to load your transformed data into a SQL database, what steps would you take to make that happen? Be sure to number steps when the order matters.

USDA ARMS Data

1. Create a table in SQL Server the follows the schema laid out in the spark dataframe
2. Coalesce this dataframe into 1 partition and write this file to an azure blob in the gold-standard container in json form
3. Create a kafka topic called 'farms'
4. Set up a producer databricks notebook and load the data file from azure
5. Create the producer object, then iterate through each row in the file, putting each column value together in a dictionary
6. Use the producer object to submit the dictionary as a message, print a success statement once the message has been sent, then have it wait 5 seconds to simulate streaming.
7. Create a kafka consumer notebook create a consumer subscribed to 'farms'
8. Set up the notebook so that every new batch of messages (latest offset) gets appended to the SQL Server table.
9. Repeat this process for the state output value data file on a separate set of producer/consumer bricks. These will run simultaneously on the same pipeline.

USDA Foreign Agriculture Export Data

1. Create a table in SQL Server the follows the schema laid out in the spark dataframe
2. Coalesce this dataframe into 1 partition and write this file to an azure blob in the gold-standard container in json form
3. Create a kafka topic called 'exports'
4. Set up a producer databricks notebook and load the data file from azure
5. Create the producer object, then iterate through each row in the file, putting each column value together in a dictionary
6. Use the producer object to submit the dictionary as a message, print a success statement once the message has been sent, then have it wait 5 seconds to simulate streaming.
7. Create a kafka consumer notebook create a consumer subscribed to 'exports'
8. Set up the notebook so that every new batch of messages (latest offset) gets appended to the SQL Server table.

Commodities API

1. Create a table in SQL Server the follows the schema laid out in the spark data frame
2. Coalesce this data frame into 1 partition and write this file to an azure blob in the gold-standard container in JSON form
3. Create a Kafka topic called 'exports'
4. Set up a producer databricks notebook and load the data file from azure
5. Create the producer object, then iterate the last row in the file, putting each column value together in a dictionary
6. Use the producer object to submit the dictionary as a message, print a success statement once the message has been sent.

7. Using a trigger in the data factory, the producer will run once every hour to pull in the updated data from the API
8. Create a kafka consumer notebook create a consumer subscribed to ‘exports’
9. Set up the notebook so that the new message (latest offset) will get appended to the SQL Server table.

Conclusion

Now that we have cleaned and transformed our data, we will be able to use this data to make predictions regarding the future prices of commodities using machine learning and examine the top producers and consumers of these commodities. We hope that this report will be useful to the producers of these commodities and other firms that work with these commodities in order to make decisions regarding their production resources. This report can also serve as a template for anyone with an interest in exploring other commodities.