ETL Project Report

Objective

This project serves as the prerequisite for the purpose of investment recommendation, projection and education. The analysis requires two different types of data which will be extracted from different sources with different methods.

1 - ETF Historical Return Data from Yahoo Finance

Summary

Yahoo finance (https://finance.yahoo.com) has a wealth of trading data on trending tickers, stocks, and ETFs. For the ETL project we have focussed on the ETF bonds. ETF data on specific tickers for a 20 year time period with a specific data frequency was extracted (web scraped), transformed and cleaned using python data frames and loaded into mySQL database.

1.1 - Data Extract

We extract, transform and load the data for Data Extract

1.1.1 Short-Period Yahoo Web Scraping Function

Scrape the historical returns based on a ticker name, a period of time (defined by start and end date in epoch format) and data frequency. The function can only extract no more than 96 sets of data at one time.

- Input:
    - Ticker Name, include Bond ETFs have both short term and long term durations, refer to *Appendix I - Treasury Bond ETF* for further details.
    - Start date, in datetime format;
    - End date, in datetime format;
    - Optional, data frequency:
        - 1d (every business day);
        - 1wk (every week);
        - 1mo (every month)
- Process
    - Convert the start date and end date to epoch and unix timestamp.
    - Config the URL that can call Yahoo data with the specific ticker name, converted start date, converted end date and data frequency with the following format:
        - url = 'https://finance.yahoo.com/quote/' + <ticker> + '/history?period1=' + <date1_epoch> + '&period2=' + <date2_epoch> + '&interval='+<frequency>+'&filter=history&frequency='+<frequency>
    - Web Scrape the data table from the website determined by the url above.
- Output: Dataframe with the following column names:
    - Open Price

- High Price
- Low Price
- Close Price
- Adj Close Price
- Volume

1.1.2 Long-Period Yahoo Web Scraping Function

Scrape the historical returns based on the ticker name, a period of time for the longer period about 20 years by looping through the short- time period loop depending on the frequency of the data.

1.1.3 Extract historical data for each ETF within the List

Call the "Long-Period Yahoo Web Scraping Function" with each ticker within the list (refer to *Appendix I - Treasury Bond ETF* for the details of the list), date period from 2000 to today, data frequency as "daily" (i.e, every business day). Save the output dataframe within a dictionary that has the key as the ticker name.

1.2 - Transformation of Data

The data that was scraped from Yahoo was pulled into a Python data dictionary. The dataset went through iterative process that cleaned and aggregated the data.

1.2.1 Clean Data

Go through each dataframe within the dictionary (from last step) and do the following:

- Rename column names to be consistent to load into mySQL database.
- Change all the data from string to float.
- Add extra column with the ticker name.
  - As searching index after aggregating all the tables together.
- Change the date string to datetime object.

The cleaned data set displays as below:

	Date	Open	High	Low	Close	Adj_Close	Volume	Ticker
0	2007-01-11	108.70	108.70	108.70	108.70	98.78	900.0	SHV
1	2007-01-16	108.76	108.76	108.74	108.74	98.81	500.0	SHV
2	2007-04-02	108.99	109.03	108.91	109.00	99.83	23000.0	SHV

1.2.2 Aggregate Data

- Aggregate all the cleaned ticker tables.

1.2.3 Format Data Index

- Change the index name as ‘id’ and increase each index value by 1
    - Since SQL primary key is formatted as integer.

1.3 - Load to Database

- Establish connection to MySQL Database
- Define Ticker Class that will build the Ticker Table in MySQL
- Load the Historical data for the given Ticker from the Dataframe to to the Ticker SQL table.

2 - Treasury Yield Data

Summary

The U.S. Treasury ensures the nation's financial security, manages the nation's debt, collects tax revenues, and issues currency, provides data on yield rates. Leveraging Quandl Database for Yield Curve (Term Structure) of Treasury Bonds, TIPS and Corporate Bonds. Our process pulls the data, transforms and stores the data for future analysis purposes.

2.1 - Data Extract:

Quandl has enabled easy access to millions of financial and economic datasets from hundreds of publishers via a single free API. Our team pulled US Treasury data via the Quandl API call, our process to extract that data was as simple as

- Importing the Quandl library
- Setting up an API key
- Requesting a get function against the "*USTREASURY/YIELD*" financial dataset that we were interested in pulling.

2.2 - Data Transformation:

The dataset provided was pulled into a Python Dataframe then went through the following transformation steps.

- Renaming of Columns to remove spaces in preparation for data load into SQL DB.
- Change the index name as ‘id’ and increase each index value by 1
    - Since SQL primary key is formatted as integer.

The formated data set is as below:

	Date	MO_1	MO_2	MO_3	MO_6	YR_1	YR_2	YR_3	YR_5	YR_7	YR_10	YR_20	YR_30

0	1990-01-02	NaN	NaN	7.83	7.89	7.81	7.87	7.90	7.87	7.98	7.94	NaN	8.00
1	1990-01-03	NaN	NaN	7.89	7.94	7.85	7.94	7.96	7.92	8.04	7.99	NaN	8.04
2	1990-01-04	NaN	NaN	7.84	7.90	7.82	7.92	7.93	7.91	8.02	7.98	NaN	8.04

Note: Not delete the column with NaN value since the further yield curve analysis is better to be based on longer historical information.

2.3 - Load to Database

- Establish a connection to MySQL Database
- Define Yield Curve Data Class that build the Yield Curve Data Table in MySQL
- Load the Yield Rates from the Dataframe to to the Treasury_Yield SQL table.

Appendix Ⅰ - Treasury Bond ETF

Table 1 - Bond ETF with Long Term Duration

Ticker	Full Name	Description	Data Frequency	Data Period
IEF	iShares 7-10 Year Treasury Bond ETF	IEF tracks a market-value-weighted index of debt issued by the US Treasury with 7-10 years to maturity remaining. Treasury STRIPS are excluded.	Daily	Jul 30, 2002 to Feb 22, 2019
DTYL	iPath US Treasury 10-year Bull ETN	DTYL tracks the Barclays 10Y U.S. Treasury Futures Targeted Exposure Index. The index gains a point for each 1 bp drop in yield and the note gains 10 cents for each 1 point gain in the index.	Daily	Aug 11, 2010 to Jan 30, 2019
EDV	Vanguard Extended Duration Treasury ETF	EDV tracks a market weighted index of high-duration zero-coupon US Treasury securities.	Daily	Jan 31, 2008 to Feb 22, 2019
TLH	iShares 10-20 Year Treasury Bond ETF	TLH tracks a market-weighted index of debt issued by the U.S. Treasury. Remaining maturity must be between 10 and 20 years.	Daily	Jan 12, 2007 to Feb 22, 2019

Table 2 - Bond ETF with Short Term Duration

Ticker	Full Name	Description	Data Frequency	Data Period
SHV	iShares Short Treasury Bond ETF	SHV tracks a market-weighted index of debt issued by the U.S. Treasury. Remaining maturity must be 1-12 months.	Daily	Jan 11, 2007 to Feb 22, 2019

VGSH	Vanguard Short-Term Treasury Index ETF	VGSH tracks a market weighted index of fixed income securities issued by the U.S. Treasury, excluding inflation-protected securities, with maturities of 1-3 years.	Daily	Dec 31, 2009 to Feb 22, 2019
SCHR	Schwab Intermediate-Term U.S. Treasury ETF	SCHR tracks a market weighted index of investment grade debt issued by the U.S. Treasury with remaining maturity of 3-10 years.	Daily	Aug 05, 2010 to Feb 22, 2019

Appendix Ⅱ - Treasury Yield Curve

Table 3 - Treasury Yield Curve

Maturity	Data Frequency	Data Period
1 Month	Daily	From 2001-07-31
2 Month	Daily	From 2018-10-16
3 Month	Daily	From 1990-01-02
6 Month	Daily	From 1990-01-02
1 Year	Daily	From 1990-01-02
2 Year	Daily	From 1990-01-02
3 Year	Daily	From 1990-01-02
5 Year	Daily	From 1990-01-02
7 Year	Daily	From 1990-01-02
10 Year	Daily	From 1990-01-02
20 Year	Daily	From 1993-10-01
30 Year	Daily	From 1990-01-02