Spring 6-4-2024

Experimental Methods in Predicting Market Drift and other Portfolio Optimization Factors using Graph Theory

Perry Harrison Zhang
Dartmouth College, perry.h.zhang.24@dartmouth.edu

Experimental Methods in Predicting Market Drift and other Portfolio Optimization Factors using Graph Theory

by

Perry H. Zhang

Professor Douglas Mcilroy

A Dissertation

Submitted to the Department of Computer Science

Dartmouth College

In Partial Fulfilment of the Requirements

For the Degree of Undergraduate High Honors in Computer Science

May 2024

Introduction

In the field of mathematical finance, models will encapsulate both deterministic factors, such as risk-free interest rate, and indeterministic factors, such as drift and volatility. In this study, I examine how prediction of foreign exchange rates between pairs of countries can be examined via graphs. Due to the limitation of data within my budget, we examine macro currencies, which is provided through an EODHD subscription. EODHD is an end of day high-definition data provider for NASDAQ and other financial institutions. In both experiments, we construct nodes representing countries that contain important macro data. Each edge between each node can be seen as a macro currency. With machine learning, we determine which of these factors influence the binary classification problem.

Experiment 1 demonstrates what factors are influential to the "binary classification" problem for predicting an uptrend in asset price from a holding or downtrend. The simple binary classification alone provides good accuracy and lower F1[1] recall and precision. This is common with binary classification problems, and shows that we are identifying factors for uptrend. The results show that relative to time period and economic health, different factors are more important than others. Quadrants are a measure of the first order derivative of Consumer Price Index and Gross Domestic Product. Depending on when CPI is bigger and smaller than 0, and when GDP is bigger or smaller than 0, we develop four economic quadrants measuring the condition of an economy. We can better capture this "context" via defining the relative economic quadrants a country is and with various lag features.

Experiment 2 improves on the features in experiment 1. Through functions that format the first and second derivative, as well as z score, moving_average, and volatility for 0, 5, 10 lag windows, we enhance 29 selected features from EODHD, contrasted to the previous 39. Additionally, we use calendar data in the form of panda dataframes[2], marked 1 on dates that hold important economic events and 0 otherwise. In regression, we see improvements through Mean Squared Error and R^2 measures.

In experiment 3 I create graphs and show how arbitrage opportunities may take the form of a highest cost cycle, where each walk's exchange rate is multiplied by the succeeding edge's exchange rate. I show how this is completed through NetworkX[3], and using cycle finding algorithms . I then examine the influence of different time periods, cycle length, and how they influence returns, and volatility.

Experiment 1

Goal: Observe performance, discover important features through weights for future machine learning

Forex Pairs: CNYUSD, EURUSD, GBPUSD, AUDUSD, CADUSD, JPYUSD

[1] F1: The harmonic mean of Precision and Recall, used to measure a test's accuracy that considers both the precision and the recall.
Accuracy: The proportion of correct predictions among the total number of cases processed. It is useful for balanced datasets.
Recall: The proportion of actual positives that were correctly identified, reflecting the model's ability to find all relevant cases.
Precision: The proportion of actual positives among the predicted positives, highlighting the correctness of positive predictions.

[2] Pandas: A python library providing a variety of datamining formatting capabilities.
Dataframe: a data structure that conveniently reads csv and other filetype data.

[3] NetworkX: A high level, python library used for creating complex graphs.

Countries: USA, GBR, JPN, AUS, CAN, FRA, CHN (For the exception of EUR, all the other currencies are represented by one country. I chose FRA to substitute as the country that we are using to estimate EUR, as France carries a hugely important economy in Europe).

Input features: 39 macro features of country A, 39 macro features of country B, exchange rate. 79 features in total.

Prediction: 1 for an increase in exchange rate, 0 for same / decrease.

Data: The 39 macro features are detailed here. The exchange rate uses a different api, but also from EODHD.

Model: One layer, 79 input Logistic Regression, default Pytorch model.

Data mining:

First, we collect all country data via our API. EODHD details macro-economic data from 1960-2022. Some countries do not have early / late data. To fill these data points, we use the linear interpolate function. For exchange rate data, the data is provided daily. We rescale the exchange rate data to be in a monthly format, and also interpolate accordingly. Because some initial data is unstable, we choose to start with index 100 when selecting any rows for training and testing.

Third, we calculate the percent change between current and previous values. We refill this entire dataframe with these values.

Training:

The lookahead is how many steps ahead we are predicting. The lookbehind is how many steps the model observes before the prediction. For the lookbehind steps, we "train" the model. When we are finally asked to predict the lookahead, we "test" the model.

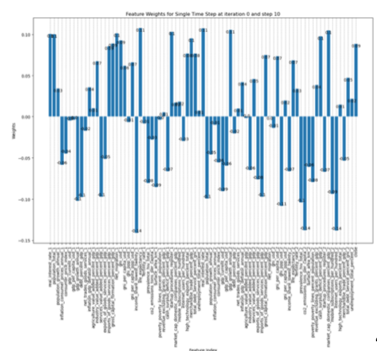
Testing:

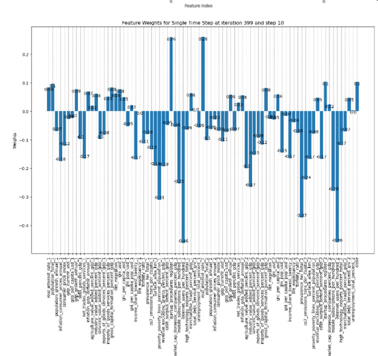
The lookahead is how many steps ahead we are predicting. The lookbehind is how many steps the model observes before the prediction. For the lookbehind steps, we "train" the model. When we are finally asked to predict the lookahead, we "test" the model.

Evaluation:

We save a training and testing log for each model. We keep the true values of the binary classification and our predictions for both the training and testing models. We can then calculate accuracy, precision, recall, F1.

CNYUSD

 [4]



4

Highest Positive Weights:

Real_interest_rate_2=0.28 USA

Market_cap_domestic_companies_percent_gdp=0.26 CHN

Startup_procedures_register_2=0.11 USA

Lowest Negative Weights:

Internet_users_per_hundred_1=-0.48 CHN

High_technology_exports_percent_total_2=-0.48 USA

Prevalence_hiv_total_2=-0.37 USA

Metrics:

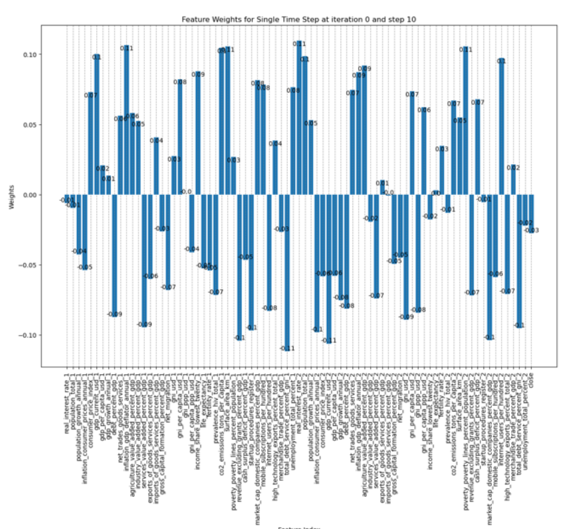
Accuracy: 0.87

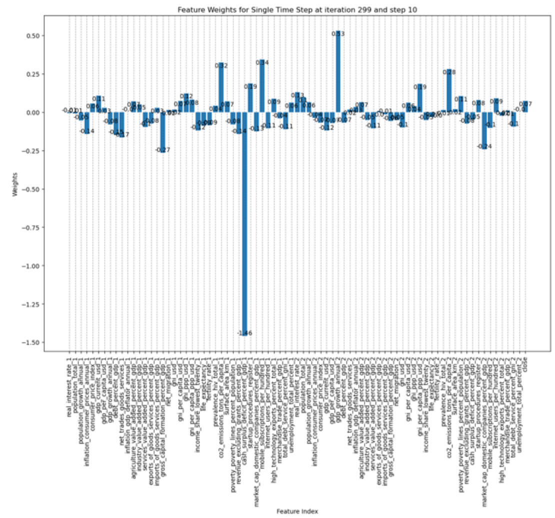
Precision: 0.0

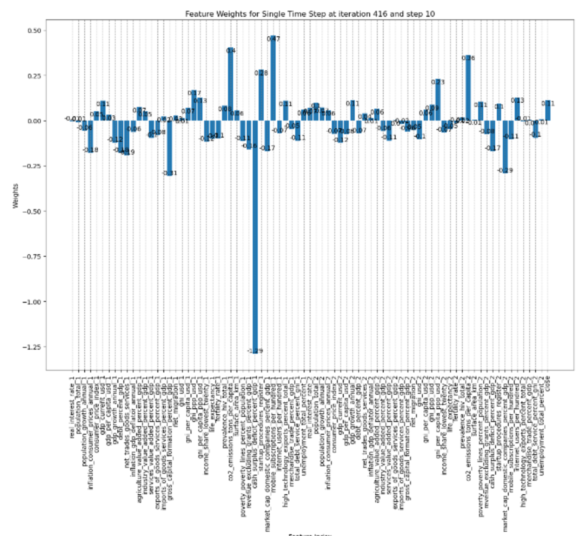
Recall: 0.0

F1 Score: 0.0

EURUSD







Highest Positive Weights:

Mobile_subscriptions_per_hundred_1 FRA

Co2_emissions_tons_per_capita_1 FRA

High_tech_exports_per_hundred_1 FRA

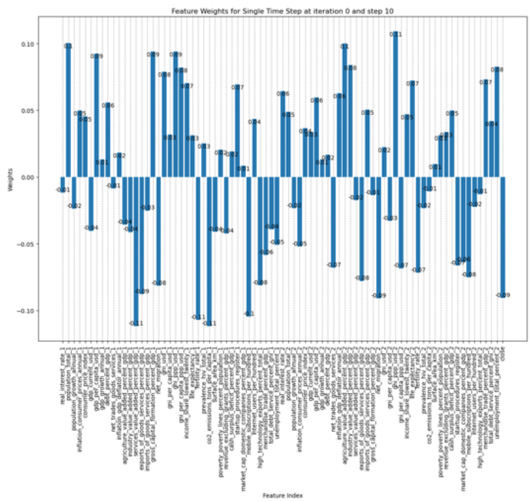
Lowest Negative Weights:

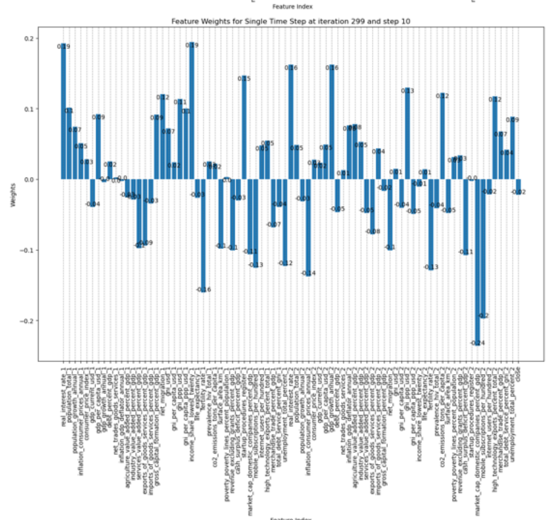
Cash_surplus_deficit_percent_gdp_1 FRA

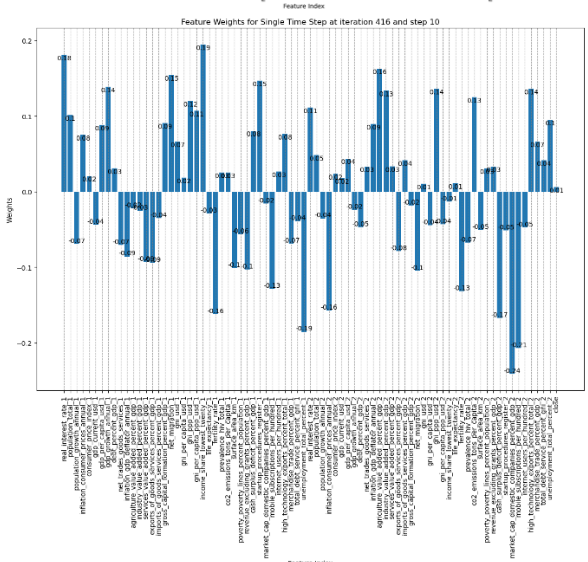
Market_cap_domestic_companies_percent_gdp_2 USA

Gross_capital_formation_percent_gdp_1 FRA

GBPUSD







Highest Positive Weights:

Income_share_lowest_twenty_1=0.19 GBR

Real_interest_rate_1=0.18 GBR

Agriculture_value_added_per_2=0.16 USA

Lowest Negative Weights:

Market_cap_domestic_companies_percent_gdp_2=-0.24 USA

Mobile_subscriptions_per_hundred_2=-0.21 USA

Unemployment_total_percent_1=-0.19 GBR

Metrics:

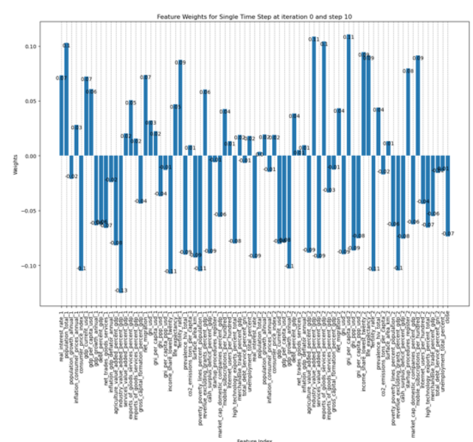
Accuracy: 0.54

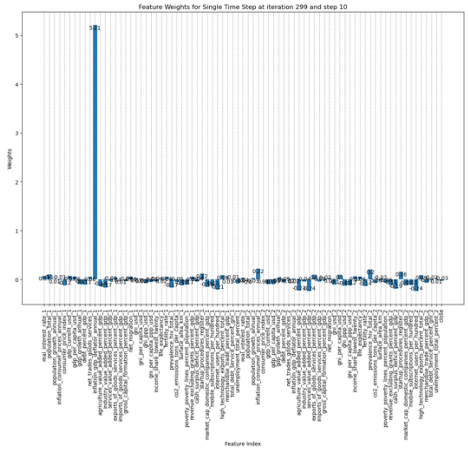
Precision: 0.43

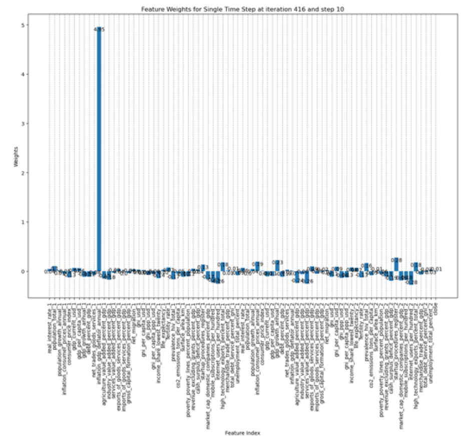
Recall: 0.05

F1 Score: 0.09

AUDUSD







Highest Positive Weights:

Inflation_gdp_deflator_annual_1=4.95 AUS

Startup_procedures_register_2=0.24 USA

gdp_growth_annual_2=0.23 USA

Lowest Negative Weights:

Internet_users_per_hundred_2=-0.28 USA

Services_added_percent_gdp_2=-0.26 USA

Internet_users_per_hundred_gdp_1=-0.24 AUS

Metrics:

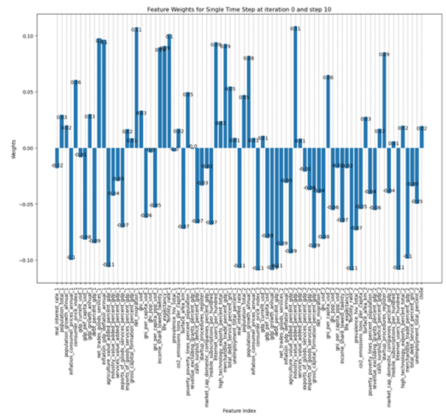
Accuracy: 0.76

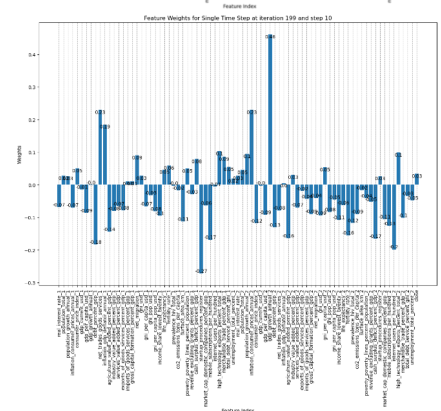
Precision: 0.38

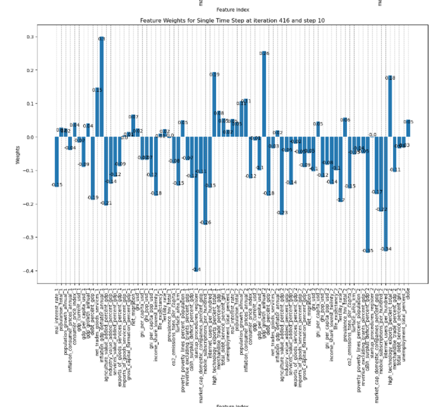
Recall: 0.05

F1 Score: 0.09

CADUSD







Highest Positive Weights:

Inflation_gdp_deflator_annual_1=0.3 CAN

Gdp_growth_annual_2=0.26 USA

High_technology_exports_percent_total_2= 0.18 CAN

Lowest Negative Weight:

Startup_procedures_register_1=-0.4 CAN

Cash_surplus_deficit_percent_gdp_2=-0.35 USA

Internet_users_per_hundred_2=-0.34 USA

Metrics:

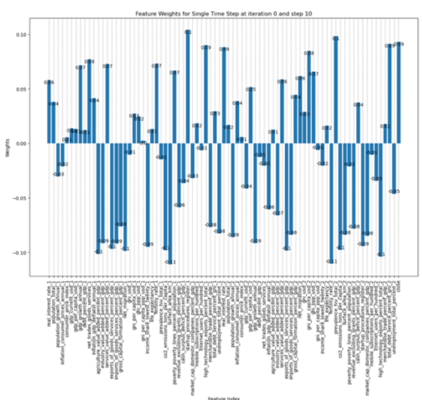
Accuracy: 0.81

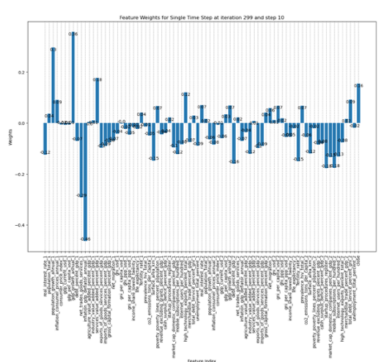
Precision: 0.63

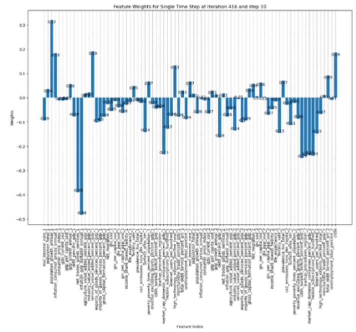
Recall: 0.04

F1 Score: 0.07

JPYUSD







Highest Positive Weights:

Population_growth_annual_1=0.32 JPY

Services_value_added_percent_gdp_1=0.19 JPY

Close_2=0.19 USA

Lowest Negative Weights:

Inflation_gdp_deflator_annual_1=-0.48 JPY

Net_trades_goods_services_1=-0.39 JPY

Cash_surplus_deficit_percent_gdp_2=-0.25 USA

Metrics:

Accuracy: 0.73

Precision: 0.28

Recall: 0.05

F1 Score: 0.08

Conclusion

Looking at the cases for most of the forex pairs, we see that there is a high accuracy. For currencies where the respective countries that seemed to be more closely tied, the precision is higher. An example of this is CAN and USA, with the CADUSD forex model. The precision is 0.6, which is promising. However, for most other pairs, recall and precision is low, most below 0.5.

Clearly, there is also a lot of noise. Both in terms of economic timing, and which features we are using. A notable

Top features:

Real_interest_rate

High_tech_exports_percent_gdp

Inflation_gdp_deflator_annual

Startup_procedures_register

Mobile_users_per_hundred

Internet_users_per_hundred

Market_cap_domestic_companies_percent_gdp

Population_growth_annual

Services_value_added_percent_gdp

Close

Net_trades_goods_services

Startup_procedures_register

We see that with the simple 2-country model, many of the countries have the above features highly influencing the exchange rate. We see that US's high_tech_export_percent_gdp greatly reduces the exchange rate in most cases. These include real_interest_rate, Inflation_gdp_deflator_annual, etc. We also notice that some features are particular to country.

JPY, for example, has a population growth problem, and deflation in currency. AUS also suffered from inflation previously. However, the high weights attributed to these features are particular to country and political policy, which makes it unique to the currency pair we are studying.

We see that the model is picking up nuances of each country, but perhaps these nuances are too general.

Limitations:

Because this is a binary prediction problem, and both "same price" and "lower price" are both grouped into the 0 case, it is natural to see a lower recall.

We are also losing a lot of temporal data with a one-layer logistic regression model. Other models should be considered, as well as other features.

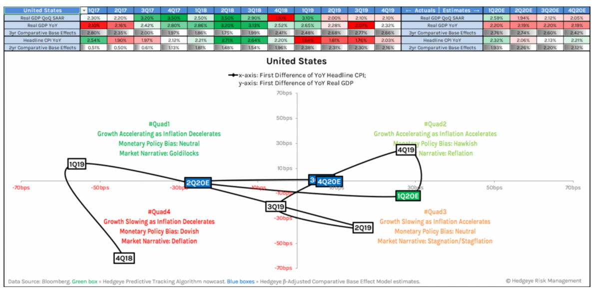
There is also a loss of scope of time. What I mean by this is that some financial health and other metrics require a grasp of how the economy is doing in terms of recession and inflation, and the economic policies and stances associated with such.

Improvements:

The first goal of this experiment is to use the important features we have discovered, and reduce the noise of the other features.

The model provides interpretability for the most important features, which we will use in future experiments. However, we may consider binomial trees, LSTMs, and other predictive algorithms. This will improve our metrics, but likely make it more difficult for readability. We will also most definitely add rolling averages, Bollinger bands, and other technical indicators to further assist our predictions.

Hedgeye[5], a proprietary research firm, often times considers a country's economic health by measure of first or second order of change for GDP and CPI. This calls for a second derivative measure of GDP that we do not account for in our first derivative training data (percentage increases).



Quad 4 Investing Playbook

Furthermore, certain countries might have a collective effect on currencies. EUR is not just used in FRA, but also DEU and other countries. This is not the biggest issue, and we will address the others first.

While we can see that certain factors do contribute to the value of a currency, it is important to realize what context they are in. This isn't readily captured with a simple binary classification problem. However, this experiment does allow insight into what factors can be important to what countries in different times.

Experiment 2: Feature Selection and Enhancement

Countries: all possible EODHD countries

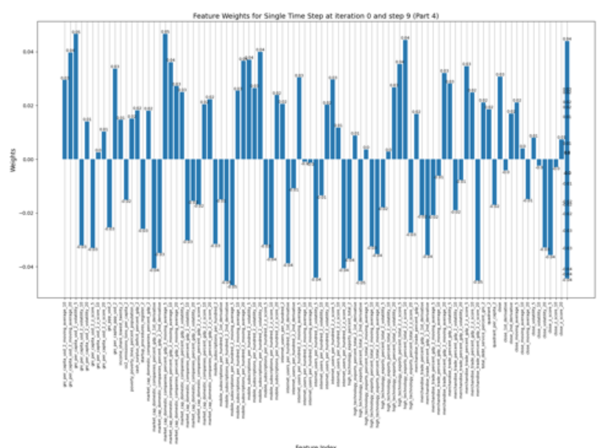
Input features: 29 macro features of country A, 29 macro features of country B, exchange rate. Among each of the 29, 18 features are enhanced, as in there are 12 features for every one feature that needs to be enhanced. Finally, there is quadrant 1 and quadrant 2, and 12 enhanced features for 'close' or exchange rate. 446 features in total.

Prediction: Regression on 'close' or exchange rate.

Data: The 39 macro features are detailed here. The exchange rate uses a different api, but also from EODHD.

Model: One layer, 446 input Logistic Regression, default Pytorch model.

To improve the performance of the model, we may incorporate temporality, add rate of change in various orders, reduce noise, and add lag. We also add important economic event dates. In this section we examine how that improves our predictions, and predict market drift through logistic regression.



Snapshot of Experiment 2 Feature Weights

Features such as temporality, here measured by the economic quadrants, are the most important, with weights of over 5% influence, out of 446.

The enhancing of certain features, such as forex price z-score, also help the

[5] Hedgeye: Proprietary investment research analysis platform

prediction. This is standardly known as the Bollinger band method. If an assets price deviates away too far from a lag mean, then a sudden rise or dip might occur to revert back to mean. In fact, we see that if the price deviates too far from the median, often after two standard deviations, the efficiency of the market brings it back to mean.

Experiment 3: Profit Maximizing Questions with graphs

In experiment 3 I demonstrate how graphing all countries and their currency pair edges can reduce profit-maximizing problems into graph problems.

A profit-maximizing cycle starting and ending at country a is defined as a cycle in which the multiplication of exchange rates (edge weights) is maximized. Because our original graph contains all such information, at each given timestep we create and analyze snapshots of the graph. Here, I have assumed that in order to exchange across markets, it will take at least one day to exchange funds. That is, if we detect an arbitrage cycle, perhaps the day after such a cycle may change. Here, the

Goal: Find different cycles for the world graph G at different prediction time periods. Compare the predicted gain against actual gain for G. Compare the effects of stable and volatile times on the predicted gain and actual gain.

Forex Pairs: all possible EODHD forex pairs

Countries: USA, GBR, JPN, AUS, CAN, FRA, CHN (For the exception of EUR, all the other currencies are represented by one country. I chose FRA to substitute as the country that we are using to estimate EUR, as France carries a hugely important economy in Europe).

Input features: 227 enhanced macro features of country A, 227 enhanced macro features of country B, 12 features for exchange rate. 446 features in total.

Regression: Regression on the exchange rate

Data: The 39 macro features are detailed here. The exchange rate uses a different api, but also from EODHD.

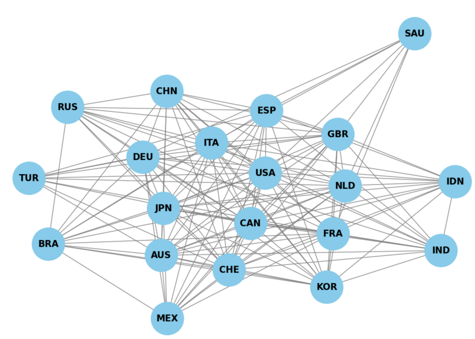
Time Period Data:

- Stable:
  - 1945-1965: post WW2
  - 1980-1990: general prosperity and developments in the world
  - 2009-2020: post 2008 housing financial crisis
- Volatile:
  - 1929-1932: first world war
  - 1970-1980: great inflation
  - 2008: 2008 housing financial crisis

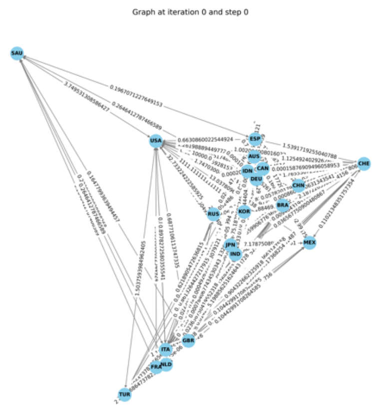
Model: One layer, 446 input Logistic Regression, default Pytorch model.

Time periods: stable

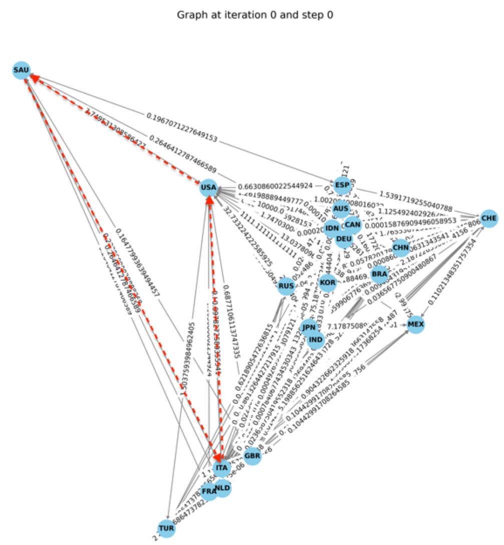
A test of this is showing the close relationship between a graph G's predicted gain matrix vs the actual profit-maximizing cycle profit.



Graph



Example Digraph at timestep[6]



Example arbitrage cycle in red

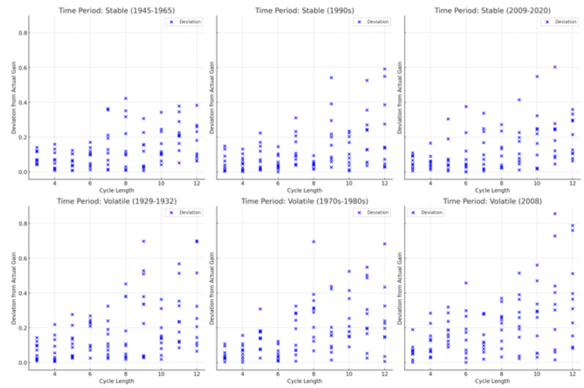
If we take USA, ITA, and SAU from this graph, we can construct a an arbitrage cycle for USA ITA SAU USA for say, three fictional exchange rates 5, 0.2, and 1.1.

$$5 * 0.2 * 1.1 = 1.1$$

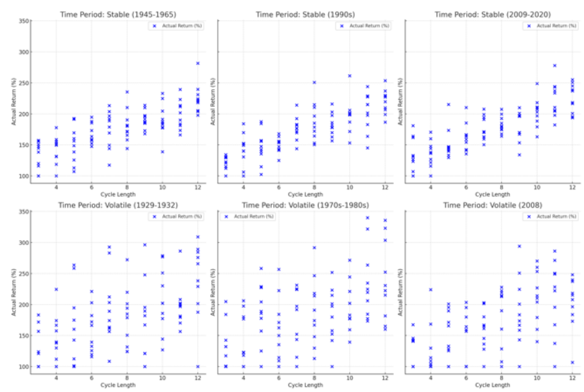
$$1.1 > 1$$

We see that the return is higher than 1, and we have found an arbitrage cycle.

Through iterations of the graphs through our training data, we may keep track of the actual cycles, the predicted cycles, and their differences from the actual cycles



Predictions and actual gains vs cycle length



Actual gains vs Cycle Length

Notice that cycle length is positively associated with returns, both actual and predicted. This is because the longer the cycle, the more exponentiation we receive from each exchange. This also comes with more risk and volatility, however. This is because while you may traverse along a cycle, there is no guarantee that the cycle won't change in the next time step.

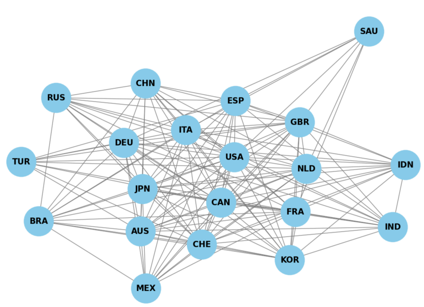
The experiment also shows that the most profitable arbitrage opportunities usually stem from volatile time periods. This is because the wider range of well operating and ill operating economies during volatile

[6] Digraph: Directed graph, a graph with directed edges

time periods enable for greater arbitrage opportunities. Factors such as heavy inflation and economic instability give currencies pairs such as JPYUSD enormous fluctuation during WWII, especially when the US and other stable countries have no change in exchange rate. Even recently, the massive fluctuations of the Turkish Lira have provided such arbitrage opportunities, some platforms such as IBKR, a leading self investor platform, even banning the trade to prevent too much.

Conclusion

We see that for cycles of the large length, there is higher range of returns. We see that for volatile time periods, there is higher gain to be made. Despite so, there are differences between actual gain and predicted gain. During the stable time periods, there is less variation between actual gain and predicted gain, contrasted to volatile time periods. A simple explanation is that there is a factor of P(2^{nd} up| 1^{st} up) that we need to consider in order to traverse the exchange cycle, since we may only traverse one edge per day. Because of this, there might be changes in the exchange rate, and the cycle itself might temporarily, or permanently lose its arbitrage. Moreover, there are further economic and social conditions that need to be considered, such as war, friendliness between countries, and efficiency in financial sector and connectivity (such as the USA).



USA clique, or well connectiveness in the graph

System Documentation

Perry Zhang

The descriptions provided match the functions detailed within the module, suggesting their usage for data manipulation and feature engineering specifically for time series forecasting tasks.

config.py

- DATABASE_DICT: It holds references to different databases by numerical identifiers, used for end-of-day historical data connections and API calls.
- DATABASE: Represents the database to be used, selected from DATABASE_DICT. Here, "2" specifies using Trading Economics.
- init_date and end_date: These are the starting and ending points for querying data ranges.
- EODHD_API_KEY and TE_API_KEY: The API keys for accessing the End of Day Historical Data and Trading Economics APIs.
- CLASSES: Relative directory path where classes or related data are stored.
- DEVICES and DEVICE: The list of computing devices available and the one being used for PyTorch machine learning, respectively.
- saved_countries, saved_currencies, saved_currency_pairs, saved_graph_screenshots, saved_models: Paths to directories for storing specific types of data like country information, currency details, and machine learning models.
- GRANULARITIES: It is a dictionary mapping integers to strings representing different time granularities such as '1W' for one week, '1M' for one month, etc.
- GRANULARITY: Specifies the chosen time granularity for data processing from the GRANULARITIES dictionary.
- COUNTRIES: A list of countries, identified by their codes, which will be included in the data processing.
- FOREX_PAIRS: Specifies the currency pairs that will be considered for the analysis.
- FEATURES_DICT: A dictionary that represents various economic indicators which are used after machine learning for easy access.
- TO_ENHANCE: A list of features from FEATURES_DICT that will undergo enhancements such as calculating derivatives and moving averages.
- COUNTRIES_CURRENCIES: A dictionary indicating which currency is used by which country.
- EODHD_MACRO_PARAMS_DICT and TE_MACRO_PARAMS_DICT: Dictionaries holding macroeconomic parameters used to fetch data for different countries from EODHD and Trading Economics respectively.

test.ipynb

- For training and testing the models

analyze.ipynb

- For analyzing the performance of the models

Important_economic_dates.csv

- A csv file used to manually determine the important economic dates for countries, for training purposes.
- Will be significantly easier with subscription to Trading Economics

.Classes package

AnalyszeUtils.py

- Utilities for analyzing models via themselves or their logs. Changes with experiment to experiment
- basic_metrics(log) will give you the Accuracy, Precision, Recall, F1 Score via a model's log, which keeps track of the predictions and the true values

CacheUtiles.py

- This script includes functions to clear saved files and directories in various categories, including countries, currencies, and graphs.
- These descriptions encapsulate the functionality within each script based on your existing code comments and structure. They provide a general understanding of each file's purpose and key operations within the machine learning and data processing pipeline you have established.

Calendar.py

- Because we are not using Trading Economics, we may need this file to help read in calendar data

ClassImports.py

- Used for importing libraries and files into all class files.
- optimized for loading purposes

Country.py

- Country class that is used to store country objects
- A class storing national data. It fetches and interpolates macroeconomic indicators from a specified database and stores them along with country-related information such as associated currencies.

CountryUtils.py

- save_countries: Saves a list of country objects as pickle files.
- load_countries: Loads a directory of pickled country objects, reinstantiating their non-picklable API attributes.
- country_exists: Checks if the saved pickle file for a country exists.
- load_country: Loads a single country object from a pickle file, reinstantiating its API attribute.

Currency.py

- Currency: A basic class that holds only the name of a currency.

CurrencyPair.py

- CurrencyPair: A class representing a currency pair, which fetches and processes exchange rate data for the pair.

CurrencyPairUtils.py

- Functions to save and load multiple currency pairs to and from pickle files.
- load_currency_pair: Loads a single currency pair object from a pickle file, reinstating its non-picklable API attribute.
-

DFUtils.py

- Miscellaneous dataframe-related functions like find_key_case_insensitive which locates a column in a DataFrame regardless of case.

- Functions for interpolating Series or DataFrames like linear_interpolate_series.

EdgeData.py

- Defines functions to create feature and prediction columns for graph edges representing forex prediction data. Each function takes macroeconomic data and exchange rates to generate a predictive dataset. EodhdClient.py
- EodhdClient: A class wrapping the EOD Historical Data API, enabling fetching of various financial data.

EodhdClientUtils.py

- Additional functionality for EodhdClient class, part of the official package.

JDModel.py

- Represents a Jump Diffusion model, simulates paths, and includes custom loss functions for error backpropagation. The class also contains methods for training and iteration. JumpMagnitudeModel.py
- The Model used to determine that the magnitude of the jump is

Jump OccurrenceModel.py

- Binary Classification model to predict when the jump is

LRModel.py

- LRModel: A logistic regression model implemented in PyTorch, used for binary classification problems.

LSTMModel.py

- This class encapsulates a Long Short-Term Memory (LSTM) model, suitable for time series data. It includes initialization of the LSTM layers, linear layers, a mean squared error loss function for regression, and the Adam optimizer.
- LSTMModel: A class for an LSTM-based model suited for time-series predictions.

NetworkXGraph.py

- This class leverages the NetworkX library to manage graphs representing interactions between countries. Methods include initialization with countries and exchange rate edges, adding machine learning models as edges, and iterating over the graph.
- A class utilizing NetworkX to create a graph representing the interactions between countries or entities based on financial data.

NetworkXGraphUtiles.py

- Utilities for the graph network, possibly including functions for graph analysis and manipulation.

PytorchModelUtils.py

- This module contains utilities for PyTorch models. Notable functions include:
- load_model: Loads a PyTorch model from a custom path given the path, feature count, and device type.

TEClient.py

- Handles connections to Trading Economics, including login with an API key and fetching indicator data in chunks.

TEClientUtils.py

- Currently empty

TrainTest.py

- Currently empty

TrainTestUtils.py

- designed to enhance training datasets and prepare them for machine learning tasks. It contains two primary functions:
- enhance_training_data(df, columns_to_enhance=TO_ENHANCE): This function is for preprocessing the given DataFrame by adding first and second derivatives, moving averages, volatility calculations, and Z-scores over specified window periods for the columns listed in TO_ENHANCE. This enhancement of features is crucial for capturing trends and volatility in time series data, which could be a significant factor in predicting future values.
- quadrant(df): This function classifies each row into one of four quadrants based on the second derivative of GDP and CPI. It adds a new column 'Quadrant' to the DataFrame that encodes this classification. This could be used to quickly identify the economic condition a country is in, which could be very insightful for macroeconomic analyses.

EODHD Features

- ‘**real_interest_rate**‘ – Real interest rate (%).
- ‘**population_total**‘ – Population, total.
- ‘**population_growth_annual**‘ – Population growth (annual %).
- ‘**inflation_consumer_prices_annual**‘ – Inflation, consumer prices (annual %).
- ‘**consumer_price_index**‘ – Consumer Price Index (2010 = 100).
- ‘**gdp_current_usd**‘ – GDP (current US$).
- ‘**gdp_per_capita_usd**‘ – GDP per capita (current US$).
- ‘**gdp_growth_annual**‘ – GDP growth (annual %).
- ‘**debt_percent_gdp**‘ – Debt in percent of GDP (annual %).
- ‘**net_trades_goods_services**‘ – Net trades in goods and services (current US$).
- ‘**inflation_gdp_deflator_annual**‘ – Inflation, GDP deflator (annual %).
- ‘**agriculture_value_added_percent_gdp**‘ – Agriculture, value added (% of GDP).
- ‘**industry_value_added_percent_gdp**‘ – Industry, value added (% of GDP).
- ‘**services_value_added_percent_gdp**‘ – Services, etc., value added (% of GDP).
- ‘**exports_of_goods_services_percent_gdp**‘ – Exports of goods and services (% of GDP).
- ‘**imports_of_goods_services_percent_gdp**‘ – Imports of goods and services (% of GDP).
- ‘**gross_capital_formation_percent_gdp**‘ – Gross capital formation (% of GDP).
- ‘**net_migration**‘ – Net migration (absolute value).
- ‘**gni_usd**‘ – Gini coefficient, Atlas method (current US$).
- ‘**gni_per_capita_usd**‘ – Gini coefficient per capita, Atlas method (current US$).
- ‘**gni_ppp_usd**‘ – Gini coefficient, PPP (current international $).
- ‘**gni_per_capita_ppp_usd**‘ – Gini coefficient per capita, PPP (current international $).
- ‘**income_share_lowest_twenty**‘ – Income share held by lowest 20% (in %).
- ‘**life_expectancy**‘ – Life expectancy at birth, total (years).
- ‘**fertility_rate**‘ – Fertility rate, total (births per woman).
- ‘**prevalence_hiv_total**‘ – Prevalence of HIV, total (% of population ages 15-49).
- ‘**co2_emissions_tons_per_capita**‘ – CO2 emissions (metric tons per capita).
- ‘**surface_area_km**‘ – Surface area (sq. km).
- ‘**poverty_poverty_lines_percent_population**‘ – Poverty headcount ratio at national poverty lines (% of population).
- ‘**revenue_excluding_grants_percent_gdp**‘ – Revenue, excluding grants (% of GDP).
- ‘**cash_surplus_deficit_percent_gdp**‘ – Cash surplus/deficit (% of GDP).
- ‘**startup_procedures_register**‘ – Start-up procedures to register a business (number).
- ‘**market_cap_domestic_companies_percent_gdp**‘ – Market capitalization of listed domestic companies (% of GDP).
- ‘**mobile_subscriptions_per_hundred**‘ – Mobile cellular subscriptions (per 100 people).

- ‘**internet_users_per_hundred**‘ – Internet users (per 100 people).
- ‘**high_technology_exports_percent_total**‘ – High-technology exports (% of manufactured exports).
- ‘**merchandise_trade_percent_gdp**‘ – Merchandise trade (% of GDP).
- ‘**total_debt_service_percent_gni**‘ – Total debt service (% of Gini coefficient).
- **‘unemployment_total_percent’** – Unemployment total (% of labor force).

Economy

- ‘**real_interest_rate**‘ – Real interest rate (%).
- ‘**inflation_consumer_prices_annual**‘ – Inflation, consumer prices (annual %).
- ‘**consumer_price_index**‘ – Consumer Price Index (2010 = 100).
- ‘**gdp_current_usd**‘ – GDP (current US$).
- ‘**gdp_per_capita_usd**‘ – GDP per capita (current US$).
- ‘**gdp_growth_annual**‘ – GDP growth (annual %).
- ‘**debt_percent_gdp**‘ – Debt in percent of GDP (annual %).
- ‘**net_trades_goods_services**‘ – Net trades in goods and services (current US$).
- ‘**inflation_gdp_deflator_annual**‘ – Inflation, GDP deflator (annual %).
- ‘**agriculture_value_added_percent_gdp**‘ – Agriculture, value added (% of GDP).
- ‘**industry_value_added_percent_gdp**‘ – Industry, value added (% of GDP).
- ‘**services_value_added_percent_gdp**‘ – Services, etc., value added (% of GDP).
- ‘**exports_of_goods_services_percent_gdp**‘ – Exports of goods and services (% of GDP).
- ‘**imports_of_goods_services_percent_gdp**‘ – Imports of goods and services (% of GDP).
- ‘**gross_capital_formation_percent_gdp**‘ – Gross capital formation (% of GDP).
- ‘**net_migration**‘ – Net migration (absolute value).

Economic health / Stability / Labor Statistics

- ‘**population_total**‘ – Population, total.
- ‘**population_growth_annual**‘ – Population growth (annual %).
- ‘**life_expectancy**‘ – Life expectancy at birth, total (years).
- ‘**fertility_rate**‘ – Fertility rate, total (births per woman).
- ‘**surface_area_km**‘ – Surface area (sq. km).
- ‘**gni_usd**‘ – Gini coefficient, Atlas method (current US$).
- ‘**gni_per_capita_usd**‘ – Gini coefficient per capita, Atlas method (current US$).
- ‘**gni_ppp_usd**‘ – Gini coefficient, PPP (current international $).
- ‘**gni_per_capita_ppp_usd**‘ – Gini coefficient per capita, PPP (current international $).
- ‘**income_share_lowest_twenty**‘ – Income share held by lowest 20% (in %).
- ‘**prevalence_hiv_total**‘ – Prevalence of HIV, total (% of population ages 15-49).
- ‘**poverty_poverty_lines_percent_population**‘ – Poverty headcount ratio at national poverty lines (% of population).

Confidence

- **‘co2_emissions_tons_per_capita‘** – CO2 emissions (metric tons per capita).
- **‘revenue_excluding_grants_percent_gdp‘** – Revenue, excluding grants (% of GDP).
- **‘cash_surplus_deficit_percent_gdp‘** – Cash surplus/deficit (% of GDP).
- **‘startup_procedures_register‘** – Start-up procedures to register a business (number).
- **‘merchandise_trade_percent_gdp‘** – Merchandise trade (% of GDP).
- **‘total_debt_service_percent_gni‘** – Total debt service (% of Gini coefficient).
- **‘unemployment_total_percent’** – Unemployment total (% of labor force).

Other

- **‘market_cap_domestic_companies_percent_gdp‘** – Market capitalization of listed domestic companies (% of GDP).
- **‘mobile_subscriptions_per_hundred‘** – Mobile cellular subscriptions (per 100 people).
- **‘internet_users_per_hundred‘** – Internet users (per 100 people).
- **‘high_technology_exports_percent_total‘** – High-technology exports (% of manufactured exports).

Databases:

EODHD

- Unicorn data services
- High definition end of day data
- Macroeconomic Indicators Data
- https://eodhd.com/financial-apis/macroeconomics-data-and-macro-indicators-api/

Trading Economics

- Finer Granularity than EODHD
- Over 500 indicators
- Very expensive
- https://tradingeconomics.com/

References

Meese, R. A. (1984). Is the sticky price assumption reasonable for exchange rate models? *Journal of International Money and Finance, 3*(2), 131-139. https://doi.org/10.1016/0261-5606(84)90001-9

EOD Historical Data. (n.d.). Macro indicators API. EODHD. Retrieved June 3, 2024, from https://eodhd.com/financial-apis/macroeconomics-data-and-macro-indicators-api

Hedgeye Risk Management LLC. (2024, May 20). Risk report: A Quad 4 investing playbook. Hedgeye. Retrieved June 3, 2024, from https://app.hedgeye.com/insights/81549-risk-report-a-quad-4-investing-playbook?type=what-the-media-missed