Multivariable Regression Algorithms on Currency Exchange Rates

David Heintz#1, Nhat Ho*2,

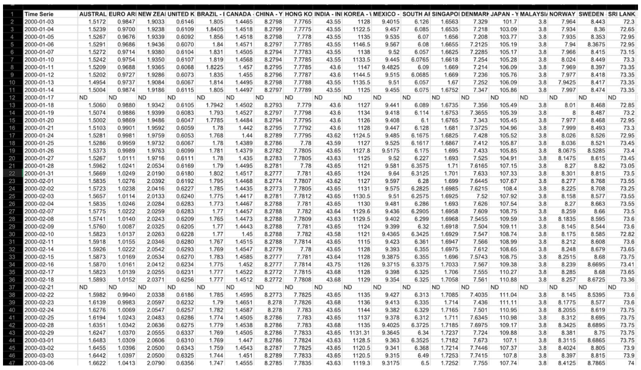
#CS596, San Diego State University
5500 Campanile Dr, San Diego, CA, United States
1dheintz@sdsu.edu
2nho0083@sdsu.edu

***Abstract*—This report explores the possibility of earlier currency exchange rate values being capable of predicting later ones. The purpose of these regression models are for use in short term trading. These models range from making predictions approximately 12 days away to predictions 100 days away. These models also consist of primarily multivariable linear regression along with 2 logistic regression models which predict whether the future value will be above or below the current one. This report will evaluate these models and examine whether they can be effectively and safely applied to currency trading. Second, this report will explore the relationships between the exchange rates of different countries and evaluate their effectiveness in being predictors for one another.**

***Keywords*— linear regression, logistic regression, exchange rate, currency, LR,USD**

I. Introduction

Currency exchanges are viable and change daily. Every country on the planet values the American Dollar(USD) at a different rate in relation to its own national currency. There are many factors which weigh into currency exchange value, some examples being stock prices, inflation, unemployment, and GDP. As shown in this dataset, the USD has a higher exchange rate in relation to some currencies (Yen), and a lower exchange rate in relation to others (Euro). While these values are ever changing, our goal is to predict the future values based on current and past values for any single country. Accurately predicting the future can be applied to currency trading and help someone make a profile by correctly guessing whether the currency exchange rate of one country with the US will go up or down. Along with this, we attempt to find relationships between multiple countries in making predictions of eachothers' exchange rate value. This relationship could allow someone to make an inference about one country's rate based on that of another.



	Time Serie	AUSTRAL	EURO AR	NEW ZEA	UNITED K	BRAZIL - I	CANADA -	CHINA - Y	HONG KO	INDIA - INI
2	2000-01-03	1.5172	0.9847	1.9033	0.6146	1.805	1.4465	8.2798	7.7765	43.55
3	2000-01-04	1.5239	0.9700	1.9238	0.6109	1.8405	1.4518	8.2799	7.7775	43.55
4	2000-01-05	1.5267	0.9676	1.9339	0.6092	1.856	1.4518	8.2798	7.778	43.55
5	2000-01-06	1.5291	0.9686	1.9436	0.6070	1.84	1.4571	8.2797	7.7785	43.55
6	2000-01-07	1.5272	0.9714	1.9380	0.6104	1.831	1.4505	8.2794	7.7783	43.55
7	2000-01-10	1.5242	0.9754	1.9350	0.6107	1.819	1.4568	8.2794	7.7785	43.55
8	2000-01-11	1.5209	0.9688	1.9365	0.6068	1.8225	1.457	8.2795	7.7785	43.6
9	2000-01-12	1.5202	0.9727	1.9286	0.6073	1.835	1.455	8.2796	7.7787	43.6
10	2000-01-13	1.4954	0.9737	1.9084	0.6067	1.814	1.4495	8.2798	7.7788	43.55
11	2000-01-14	1.5004	0.9874	1.9186	0.6115	1.805	1.4497	8.2797	7.7789	43.55
12	2000-01-17	ND	ND	ND	ND	ND	ND	ND	ND	ND
13	2000-01-18	1.5060	0.9880	1.9342	0.6105	1.7942	1.4502	8.2793	7.779	43.6
14	2000-01-19	1.5074	0.9886	1.9399	0.6083	1.793	1.4527	8.2797	7.7798	43.6
15	2000-01-20	1.5002	0.9869	1.9486	0.6047	1.7785	1.4484	8.2794	7.7795	43.6
16	2000-01-21	1.5103	0.9901	1.9592	0.6059	1.78	1.442	8.2795	7.7792	43.6
17	2000-01-24	1.5281	0.9981	1.9759	0.6053	1.768	1.44	8.2789	7.7795	43.62
18	2000-01-25	1.5286	0.9959	1.9732	0.6067	1.78	1.4389	8.2786	7.78	43.59
19	2000-01-26	1.5373	0.9989	1.9763	0.6099	1.781	1.4379	8.2782	7.7805	43.65
20	2000-01-27	1.5267	1.0111	1.9716	0.6111	1.78	1.435	8.2783	7.7805	43.63
21	2000-01-28	1.5962	1.0241	2.0534	0.6169	1.79	1.4495	8.2781	7.78	43.65
22	2000-01-31	1.5669	1.0249	2.0190	0.6180	1.802	1.4517	8.2777	7.781	43.65
23	2000-02-01	1.5835	1.0276	2.0392	0.6192	1.795	1.4468	8.2774	7.7807	43.62
24	2000-02-02	1.5723	1.0238	2.0416	0.6227	1.785	1.4435	8.2773	7.7805	43.65
25	2000-02-03	1.5657	1.0114	2.0133	0.6240	1.775	1.4417	8.2781	7.7812	43.65
26	2000-02-04	1.5835	1.0246	2.0284	0.6283	1.773	1.4467	8.2788	7.781	43.65
27	2000-02-07	1.5775	1.0222	2.0259	0.6283	1.77	1.4457	8.2788	7.782	43.64
28	2000-02-08	1.5741	1.0140	2.0243	0.6209	1.765	1.4473	8.2788	7.7809	43.63
29	2000-02-09	1.5760	1.0087	2.0325	0.6205	1.77	1.4443	8.2788	7.781	43.65
30	2000-02-10	1.5823	1.0137	2.0263	0.6228	1.77	1.45	8.2788	7.782	43.58
31	2000-02-11	1.5918	1.0155	2.0346	0.6280	1.767	1.4515	8.2788	7.7814	43.65
32	2000-02-14	1.5926	1.0222	2.0542	0.6293	1.769	1.4547	8.2779	7.78	43.65
33	2000-02-15	1.5873	1.0169	2.0534	0.6270	1.783	1.4585	8.2777	7.781	43.64
34	2000-02-16	1.5870	1.0161	2.0412	0.6234	1.775	1.452	8.2777	7.7814	43.75
35	2000-02-17	1.5823	1.0139	2.0255	0.6231	1.777	1.4522	8.2772	7.7815	43.68
36	2000-02-18	1.5893	1.0152	2.0371	0.6256	1.777	1.4512	8.2772	7.7808	43.68
37	2000-02-21	ND	ND	ND	ND	ND	ND	ND	ND	ND
38	2000-02-22	1.5982	0.9940	2.0338	0.6186	1.785	1.4595	8.2773	7.7825	43.65
39	2000-02-23	1.6139	0.9983	2.0597	0.6232	1.79	1.4651	8.278	7.7826	43.68
40	2000-02-24	1.6276	1.0069	2.0547	0.6257	1.782	1.4587	8.278	7.783	43.65
41	2000-02-25	1.6194	1.0243	2.0483	0.6286	1.774	1.4505	8.2786	7.783	43.65
42	2000-02-28	1.6351	1.0342	2.0636	0.6275	1.779	1.4538	8.2786	7.783	43.68
43	2000-02-29	1.6247	1.0370	2.0555	0.6337	1.769	1.4505	8.2786	7.7833	43.65
44	2000-03-01	1.6483	1.0309	2.0606	0.6310	1.769	1.447	8.2786	7.7824	43.63
45	2000-03-02	1.6455	1.0396	2.0500	0.6343	1.759	1.4543	8.2787	7.7825	43.65
46	2000-03-03	1.6442	1.0397	2.0500	0.6325	1.744	1.451	8.2789	7.7833	43.65
47	2000-03-06	1.6622	1.0413	2.0790	0.6356	1.747	1.4555	8.2785	7.7835	43.63

	Time Serie	KOREA -	MEXICO -	SOUTH AI	SINGAPOI	DENMARI	JAPAN - Y	MALAYSI	NORWAY	SWEDEN -	SRI LANK
2	2000-01-03	1128	9.4015	6.126	1.6563	7.329	101.7	3.8	7.964	8.443	72.3
3	2000-01-04	1122.5	9.457	6.085	1.6535	7.218	103.09	3.8	7.934	8.36	72.65
4	2000-01-05	1135	9.535	6.07	1.656	7.208	103.77	3.8	7.935	8.353	72.95
5	2000-01-06	1146.5	9.567	6.08	1.6655	7.2125	105.19	3.8	7.94	8.3675	72.95
6	2000-01-07	1138	9.52	6.057	1.6625	7.2285	105.17	3.8	7.966	8.415	73.15
7	2000-01-10	1133.5	9.445	6.0765	1.6618	7.254	105.28	3.8	8.024	8.449	73.3
8	2000-01-11	1147	9.4825	6.09	1.669	7.214	106.09	3.8	7.969	8.397	73.35
9	2000-01-12	1144.5	9.515	6.0685	1.669	7.236	105.76	3.8	7.977	8.418	73.35
10	2000-01-13	1135.5	9.51	6.057	1.67	7.252	106.09	3.8	7.9425	8.417	73.35
11	2000-01-14	1125	9.455	6.075	1.6752	7.347	105.86	3.8	7.997	8.474	73.35
12	2000-01-17	ND	ND	ND	ND	ND	ND	ND	ND	ND	ND
13	2000-01-18	1127	9.441	6.089	1.6735	7.356	105.49	3.8	8.01	8.468	72.85
14	2000-01-19	1134	9.418	6.114	1.6753	7.3655	105.39	3.8	8	8.487	73.2
15	2000-01-20	1126	9.408	6.1	1.6765	7.343	105.45	3.8	7.977	8.468	72.95
16	2000-01-21	1128	9.447	6.128	1.681	7.3725	104.96	3.8	7.999	8.493	73.3
17	2000-01-24	1124.5	9.485	6.1675	1.6825	7.428	105.52	3.8	8.026	8.526	72.95
18	2000-01-25	1127	9.525	6.1617	1.6867	7.412	105.87	3.8	8.036	8.521	73.45
19	2000-01-26	1127.8	9.5175	6.175	1.695	7.433	105.85	3.8	8.0675	8.5285	73.4
20	2000-01-27	1125	9.52	6.227	1.693	7.525	104.91	3.8	8.1475	8.615	73.45
21	2000-01-28	1121	9.581	6.3575	1.71	7.6165	107.15	3.8	8.27	8.82	73.05
22	2000-01-31	1124	9.64	6.3125	1.701	7.633	107.33	3.8	8.301	8.815	73.5
23	2000-02-01	1127	9.597	6.28	1.699	7.6445	107.67	3.8	8.277	8.768	73.55
24	2000-02-02	1131	9.575	6.2825	1.6985	7.6215	108.4	3.8	8.225	8.708	73.25
25	2000-02-03	1130.5	9.51	6.2575	1.6925	7.52	107.92	3.8	8.158	8.577	73.55
26	2000-02-04	1130	9.481	6.286	1.693	7.626	107.54	3.8	8.27	8.663	73.55
27	2000-02-07	1129.6	9.436	6.2905	1.6958	7.609	108.75	3.8	8.259	8.66	73.5
28	2000-02-08	1129.5	9.402	6.299	1.6968	7.5455	109.59	3.8	8.1835	8.595	73.6
29	2000-02-09	1124	9.399	6.32	1.6918	7.504	109.11	3.8	8.145	8.544	73.6
30	2000-02-10	1121	9.4365	6.3425	1.6929	7.547	108.74	3.8	8.175	8.585	72.82
31	2000-02-11	1115	9.423	6.361	1.6947	7.566	108.99	3.8	8.212	8.608	73.6
32	2000-02-14	1128	9.393	6.355	1.6975	7.612	108.65	3.8	8.248	8.679	73.65
33	2000-02-15	1128	9.3875	6.355	1.696	7.5743	108.75	3.8	8.2515	8.68	73.75
34	2000-02-16	1126	9.3715	6.3375	1.7033	7.567	109.38	3.8	8.239	8.6695	73.41
35	2000-02-17	1128	9.398	6.325	1.706	7.555	110.27	3.8	8.285	8.68	73.65
36	2000-02-18	1129	9.354	6.325	1.7058	7.561	110.88	3.8	8.257	8.6725	73.36
37	2000-02-21	ND	ND	ND	ND	ND	ND	ND	ND	ND	ND
38	2000-02-22	1135	9.427	6.313	1.7085	7.4035	111.04	3.8	8.145	8.5395	73.6
39	2000-02-23	1136	9.413	6.335	1.714	7.436	111.11	3.8	8.1775	8.577	73.6
40	2000-02-24	1144	9.382	6.329	1.7165	7.501	110.95	3.8	8.2055	8.619	73.75
41	2000-02-25	1137	9.378	6.312	1.711	7.6345	110.98	3.8	8.312	8.695	73.75
42	2000-02-28	1135	9.4025	6.3725	1.7185	7.6975	109.17	3.8	8.3425	8.6895	73.75
43	2000-02-29	1131.31	9.3645	6.34	1.7237	7.724	109.88	3.8	8.381	8.75	73.75
44	2000-03-01	1128.5	9.363	6.3525	1.7182	7.673	107.1	3.8	8.3115	8.6865	73.75
45	2000-03-02	1120.5	9.341	6.368	1.7214	7.7446	107.37	3.8	8.4024	8.805	73.9
46	2000-03-03	1120.5	9.315	6.49	1.7253	7.7415	107.8	3.8	8.397	8.815	73.9
47	2000-03-06	1119.3	9.3175	6.5	1.7252	7.755	107.74	3.8	8.4125	8.7865	74

Fig. 1 Foreign Currency Exchange Rate Dataset

A. Challenges

1. Since the currency exchange market is not open on holidays, there were "ND" points in the dataset which needed to be accounted for.
2. Since the market is also not open on weekends, those dates were missing from the dataset, which meant we could not attempt to use numeric assumptions of year and month lengths.
3. We first attempted a prediction of the last value of a year based on the averages of

the first 9 months. However, this proved unsuccessful with too few data points and was at risk of having a time period bias.

4. We also attempted to use a country's GDP as a factor for their respective currencies rates but through many trials/experiments we decided to move in another direction.

B. Solutions

1. To eliminate the "ND" values in the dataset we used either the drop/dropna method, or converted the dataframe to numpy arrays and used the delete method on any indices with "ND".
2. Since we could not use numerical values to separate time periods, we had to separate years and months by splitting the date data points and taking the month value of each one.
3. In order to eliminate the causes for error in the first test, we decided to split the data by months or days to add more data points for a successful regression. We also shuffled each dataset when splitting into training and testing to eliminate the possibility of time period bias. (2000 - 2010 not relating to 2010- 2019).

II. Data Preparation

For each of the future value prediction methods we needed to reformat the data into predictor (X) values and prediction (Y) values.

First, when separating the dataset by months, we used the dates dataset to help gather the first 10 days of the month to generate a predictor values of the first 5 day average and the next 5 day average. Next, we took the final value of the month and used that as the prediction value. This allowed for 240 data points in the generated set.

Second, when separating the dataset bimonthly, we followed a similar format and stored the first 15 days of the month into 3 predictor values (first 5 average, next 5 average, last 5 average). We then took the final value of the next month and used it as the predicted value. This allowed for 239 data points in the generated set.

Last, when separating the dataset to be able to make predictions of 100 days in the future based on the past 100 days, we took the started at the beginning of the dataset and took the current value (first day), the value on the 50th day, the value on the 100th day, and the value on the 200th day. This way we could use the first 3 values to predict the fourth. This had the most data points as it only removed 200 points from the initial dataset and ended up with 4819 points.

For the country relationship prediction method, we did not need to reformat the data aside from removing the ND values. This is because each of the datasets were already aligned in the initial large dataframe.

III. Model Implementation

For the time prediction models for a single country's currency, the currencies used were : Euro, Yen, Yuan, Rupee, and Australian Dollar.

For the relationship prediction models between multiple currencies, the currencies used were Canadian Dollar, Won, Yen, Singapore Dollar, and Australian Dollar.

For the first group of models, the data for a single country had to be iterated over to generate combinations of x and y values to then split into training and testing datasets.

For the second group of models the datasets for a group of countries made up the x values datasets while a single country became the y value dataset.

A. Monthly Linear Regression

1. *Training Parameters:*
    - Average of first 5 days (x)
    - Average of next 5 days (x)
    - Final day of month (y)
2. *Testing Model*

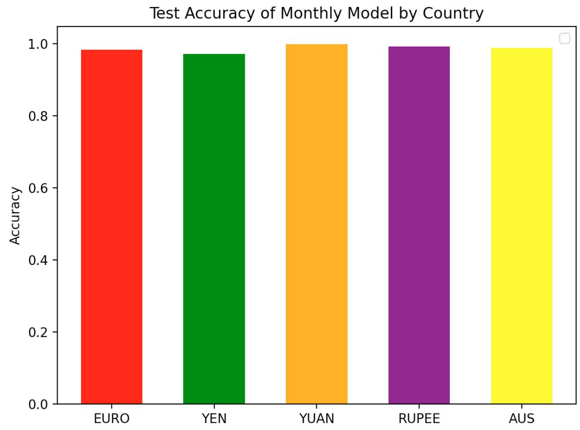


Fig. 2. Accuracy of Monthly Regression Model on 5 Countries

3. *Observations:*
   - Model achieved very high accuracy with above .97 on all countries each test, and above .99 on Yuan, Rupee, and AUS $.
   - When examining the graphs plotting the prediction and true values, none of the countries showed any bias.
   - We observed the relationship between the 10th day value and the final value for each prediction to see if their variance was higher than that of the prediction and true values.

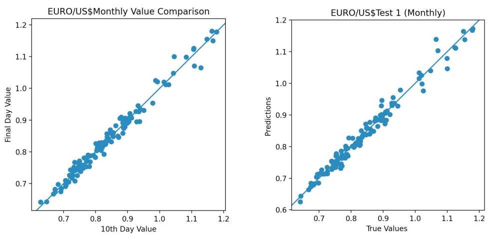


Fig. 3. Monthly Model: 10th Day/Final Day Value Comparison, Predicted/True Value Comparison

   - This similarity between the first and second graph for each of the countries proves that this monthly model doesn't have a high enough reward for it to be worth even the small amount of error the model has.
   - Although the model may predict the future value with high accuracy, since the future value is very close to the present value, that slight inaccuracy is likely to be equal to or greater than the difference between the present and future value. This means that this model will not be useful for trading.

B. *Bimonthly Linear Regression*

1. *Training Parameters*
   - Average of first 5 days (x)
   - Average of next 5 days (x)
   - Average of third 5 days (x)
   - Final day of next month (y)
2. *Testing Model*

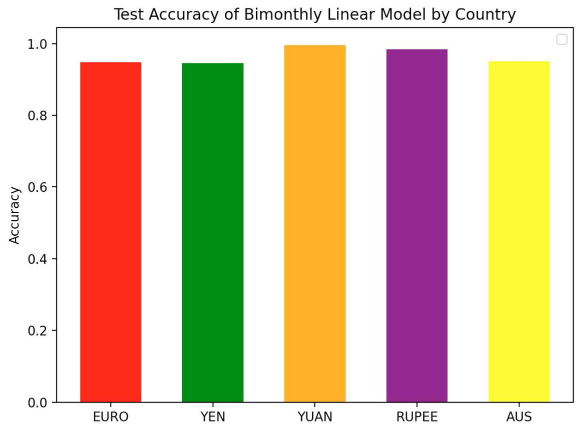


Fig. 4. Accuracy of Bimonthly Linear Model on 5 Countries

3. *Observations*
   - Model achieved high accuracy near that of the monthly model. Accuracy values ranged between .93 and .99 on all 5 models.
   - The graphs once again showed no bias, and we observed the same relationship between the predicted/true and the 15th/next month final values in order to determine whether this method was capable of being used for predicting trade outcomes.

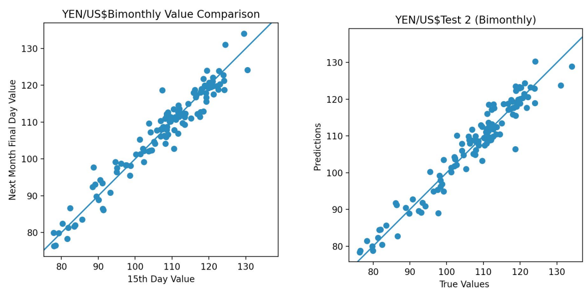


Fig. 5. Bimonthly Model: 15th Day/Next Month Final Value Comparison, Predicted/True Value Comparison

- Once again, the graphs are very similar, which shows that there is no noticeable difference between the average reward:

  $\sum |Final\ Day\ - 15th\ Day|$,

  and the average risk/error:

  $\sum |True\ -\ Predicted|$

- This shows that once again, although this model has a high accuracy, the time period is not long enough to have a high probability of reward.
- In an attempt to solve our errors, we will next use the 200 day model to try and widen the gap between final x value and y value.

C. 200 Day Period Linear Regression

1. Training Parameters

- Value of first day (x)
- Value of 50th day (x)
- Value of 100th day (x)
- Value of 200th day (y)

2. Testing Model

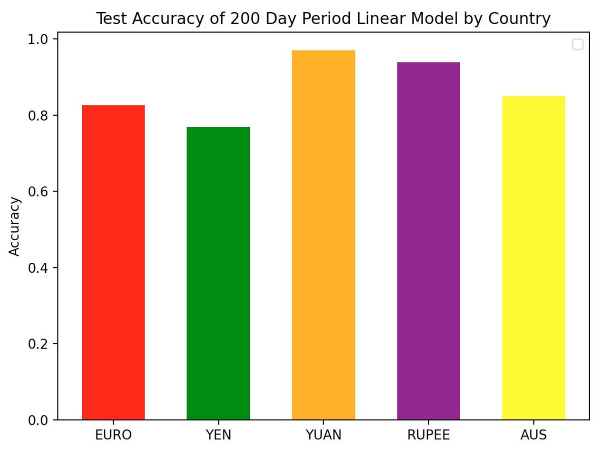


Fig. 6. Accuracy of 200 Day Period Linear Model on 5 Countries

3. Observations

- This model shows high accuracy, ranging from .75 to .95. This is less high than the previous 2 models. However that is to be expected due to the range of 100 days between x and y. This model had the largest training and testing datasets which was certainly a factor in the high accuracy.
- As depicted in the next figure by the Yuan currency, in the graphs of 100 day/200 day values and of predicted/true, we observed a bias on both the datasets of the Yuan and the AUS $.

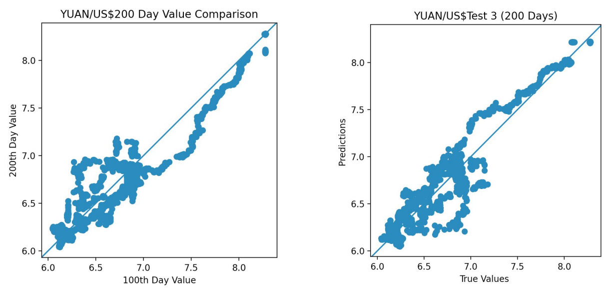


Fig. 7. 200 Day Period Model: 100th Day/200th Day Value Comparison, Predicted/True Value Comparison

- From these graphs we were able to make 2 inferences about the model and the dataset.
- First, the dataset was biased towards the future value being lower than the current value if both were high. This shows that if the currency value is high, it's predicted that it will drop. Similarly the low current values will either drop slightly, or raise largely in the next 100 days.
- The second inference we can make on this graph, and the others of this model, is that the Prediction/True value graph follows the regression line closer than the other graph. That means that the prediction error is smaller than the difference between the current and future value (reward). That means further that this model, unlike the others, has a likelihood of being applicable to trading with a positive result.

- To test this theory, we generated logistic regression models for the bimonthly method and the 200 day period method and observed if they would make or lose money from the test dataset predictions.

D. Bimonthly Logistic Regression

1. Training Parameters

- Same x values as the bimonthly linear model
- Logistic = 1 if month 2 final value > 15th day value. Logistic = 0 if month 2 final value < 15th day value

2. Testing Model

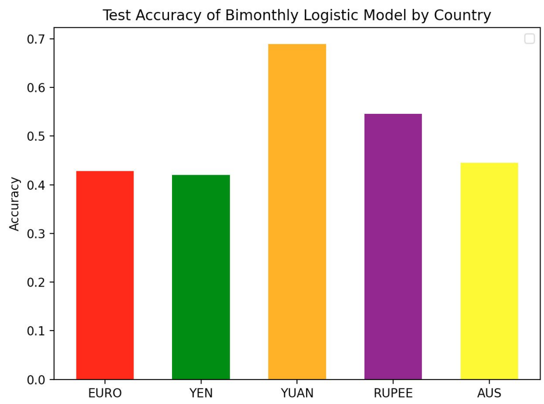


Fig. 8. Accuracy of Bimonthly d Logistic Model on 5 Countries

3. Observations

- On the average test, for this bimonthly regression model, we observed an accuracy of .5. This means that the prediction was just as accurate as if it were made entirely randomly.
- Next, we calculated the total value gained on correct predictions minus the total value lost on incorrect ones, and the results came out equally negative and positive.
- For example, in one test, the 5 countries had a highest gain of [0.9272] for the Yuan, and a largest loss of [-20.4] for the Yen. This proves that the model is not reliable.
- Furthermore, the confusion matrices were randomly spread among the 4 possibilities as well.
- The only instance which differed was the confusion matrix of the Yuan currency which frequently had 0 false or true negatives. This means that it predicted a rise in the currency every time. This ties back to the bias shown in the Yuan currency previously.

E. 200 Day Period Logistic regression

1. Training Parameters

- Same x values as the 200 day period linear model
- Logistic = 1 if 200th day value > 100th day value. Logistic = 0 if 200th day value < 100th day value

2. Testing Model

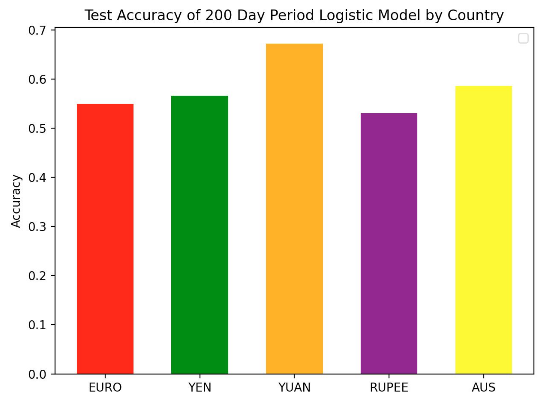


Fig. 9. Accuracy of 200 Day Period Logistic Model on 5 Countries

3. Observations

- This model, although not nearly as high in accuracy as the linear regression models, proved much more effective than the previous. As depicted above, all accuracies are slightly over 50%.
- Furthermore, when we calculated the total gain minus total loss, the values were positive for all countries on every single run. In one random run we had a highest gain of [2230.34] for the Yen, and lowest gain of

[24.302] for the Euro. This means that the model can be used effectively to guarantee making money over the testing model conditions with enough data points.
- Therefore, this model has proven that although it isn't much higher than 50% of success, it will normalize and have a positive outcome given enough trials.

F. *Five Countries to One Method:*

a) Data set description: Using a trimmed data set from the original data set we started with we were able to predict the currency of CAD with an average accuracy of 93%. We trimed the amount of counties to Australia, Canada, Korea, Japan, Singapore and Norway.

b) Evaluation metrics:

Using Linear regression, we collected data from the 5 countries and trained the data to see if we could predict the values of Canada on each date.

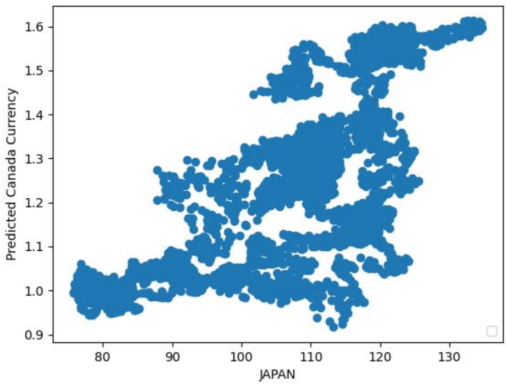


Fig. 10. Relationship Between Japanese Currency and Predicted Canadian Currency

c)Major results: Linear regression had the best result in accuray at 93% average as compared to Lasso regression which had an average of 54% accuracy.

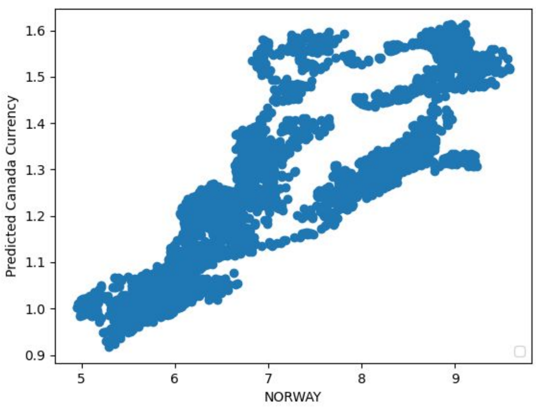


Fig. 11. Relationship Between Norwegian Currency and Predicted Canadian Currency

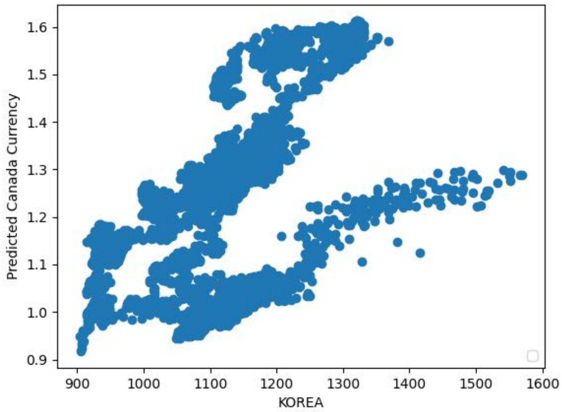


Fig. 12. Relationship Between Korean Currency and Predicted Canadian Currency

d) Analysis: The accuracy had a consistent precision for all currencies. When there is such a large data set with all similar values, the predictions make sense and we are able to train and predict what the values would be on the dates.

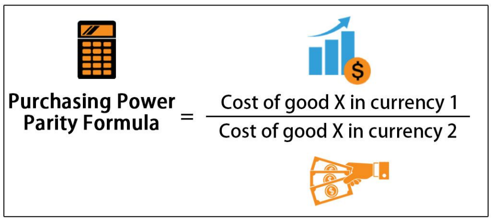
1) *Purchasing power parities :*

a)Data set description:We wanted to test the Purchasing Power Parity (PPP) as this model is often related to the idea of predicting the fluctuations of different countries's currency. In our model we attempted to use the price of crude oil and try to combine it with the currency values in USD with Canada and Australia. We originally tried comparing Mexico with Canada but the difference in Pesos compared to CAD in USD were too far apart.

	A	B	C	D	E	F	G
1	Date	Sales	Crude oil prices			AUSTRALI	EURO
2	2018/02/01	21199	65.8		2018-02-0	1.2456	0.8
3	2018/02/02	10634	65.45		2018-02-0	1.2612	0.8
4	2018/02/05	9497	64.15		2018-02-0	1.2617	0.8
5	2018/02/06	8207	63.39		2018-02-0	1.2695	0.8
6	2018/02/07	7581	61.79		2018-02-0	1.2752	0.8
7	2018/02/08	7471	61.15		2018-02-0	1.2839	0.8
8	2018/02/09	7878	59.2		2018-02-0	1.2806	0.8
9	2018/03/01	11460.7	60.99		2018-03-0	1.2883	0.8

Fig. 13. Currency Exchange Rate Dataset with Parallel Crude Oil Prices and Grocery Store Sales Values

2)Evaluation metrics:



Using linear regression we got an average of 60% accuracy with these variables. When attempting lasso progression Lasso progression the accuracy was extremely low, the negative percentile of 20%.

$$Cost\ (W) = RSS(W) + \lambda * (sum\ of\ absolute\ value\ of\ weights)$$

$$= \sum_{i=1}^{N} \left\{ y_i - \sum_{j=0}^{M} w_j\, x_{ij} \right\}^2 + \lambda \sum_{j=0}^{M} |w_j|$$

c) Major results: From our experiments, we acknowledge there is a relationship with consumer goods and the countries Currency rates.

d) d) Analysis: Incorporating a consumer good price in a model is not sufficient to create an accurate prediction of a country's currency rate. Our model only works decently when a country's USD rate is similar to another such as Canada and Australia. When we tested this model to Mexico where the peso worth was much lower, we had a low rate of predictions. The economic theory of the purchasing power parities is interesting but when incorporated in a regression it did not yield a result we wanted.

IV. Conclusions

Currency exchange has a lot of variables in play when it comes to creating a realistic prediction. We were able to experiment with some possible solutions to examine relationships between countries and between earlier and later points within each dataset. We used multiple machine learning regression techniques to generate models and determine whether or not they were effective. Although most of the models had high accuracy, few were able to provide useful and accurate information for someone intending to use this information for trading. That being said, there was much information we were able to learn from these tests, and there is still much further we can go with this project.

Nhat would like to use what he learnt from this experiment and try to create a model on ingame currencies. There are many games such as World of Warcraft and Albion online that have their own built in currency. Being able to predict the prices of items in this will be a fun project and there are less variables in game as compared to the entire world's economy.

David would like to implement his prediction models stock market datasets and attempt to predict their future values as well as make recommendations based on the logistic regression model. Furthermore, attempting to use the 200 Day Period Logistic Regression Model on actual currency datasets and examining whether or not it has the capability to produce positive outcomes over time would be an interesting way to further examine the validity of this method.

I. *References*

Datasets Used:

[1] https://www.kaggle.com/brunotly/foreign-exchange-rates-per-dollar-20002019

[2] https://www.investopedia.com/articles/forex/11/4-ways-to-forecast-exchange-rates.asp

[3] https://www.kaggle.com/rockbottom73/crude-oil-prices

Packages Installed:

[4] John D. Hunter. **Matplotlib: A 2D Graphics Environment**, Computing in Science & Engineering, **9**, 90-95 (2007), DOI:10.1109/MCSE.2007.55 (publisher link)

[5] Wes McKinney. **Data Structures for Statistical Computing in Python**, Proceedings of the 9th Python in Science Conference, 51-56 (2010) (publisher link)

[6] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. *Array programming with NumPy*. Nature 585, 357–362 (2020). DOI: 0.1038/s41586-020-2649-2. (Publisher link).

[7] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. **Scikit-learn: Machine Learning in Python**, Journal of Machine Learning Research, **12**, 2825-2830 (2011) (publisher link)

[8] Van Rossum G. The Python Library Reference, release 3.8.2. Python Software Foundation; 2020.