Milestone II

SIADS 696

Team Number 2

Final Report

Introduction

Foreign exchange markets are heavily influenced by dominant “anchor” currencies such as Euro, Yen, or Yuan. Smaller market currencies at times display a shift or dependency that are not formally quantified. Most models might use a fixed region like USD as the universal reference, but this also obscures secondary influence networks. What we are trying to discover and quantify is, currency relationships between exchange rate returns of anchoring and neighboring countries using supervised and unsupervised models. Identifying which currencies behave as a dependent of a specific anchor and measuring those relationships will provide insights for forecasting risk and optimizing cross currency exposure for traders, portfolio managers, or even policymakers.

We were motivated by a gap in FX research. FX research typically centers on USD relationships, leaving non-dollar anchor dynamics unstudied, specifically groups grouped based on economic regions comparatively similar based on behavioral characteristics.

To evaluate whether anchor neighbor currency relationships improve foreign exchange return forecasting, we implemented three supervised learning models. Linear Regression served as a transparent parametric baseline to capture simple lag relationships and to measure whether adding anchor features yields incremental linear predictability. K-Nearest-Neighbors (KNN) offered an instance base alternative that predicts future returns using local similarity in lagged and moving average feature space. Lastly, XGBoost, a tree based gradient boosting model, was included to capture nonlinear interactions and cross series dependencies that linear methods may miss. Each model was trained under two parallel conditions—with anchor and without anchor features—which allow us to directly review the marginal contribution of anchor driven information. Performance was assessed using Root Mean Squared Error (RMSE). We chose to assess performance with RMSE as it penalizes errors while returning scores in the same units as the training/test data.

We used Principal Components Analysis (PCA) to reduce dimensionality in standardized FX log returns and uncover latent co-movement structure across currencies. Then, we applied K-Means clustering to group currencies based on their similarity in PCA space. This allowed us to identify blocs of followers around anchors like EUR, CNY, and JPY without imposing predefined regional groups. ***Unlike typical FX studies that fix USD as the benchmark, our approach lets anchor influence patterns emerge from historical behavior.***

Most FX prediction studies rely on a fixed benchmark currency. In contrast, our frame integrates supervised forecasting with anchor agnostic design, testing predictive accuracy changes when cross series features are included or excluded. By applying the same with/without anchor setup across three fundamentally different learning families, LR, KNN, and XGBoost, we isolate when and how anchor currencies contribute to return predictability. Combined with unsupervised PCA and K-Means clustering used to identify potential currency, our approach moves beyond fixed regional or USD centric assumptions.

Our supervised results show that anchor features did not consistently improve predictive performance across any of the three model families. In XGBoost, none of the evaluated currency pairs achieved lower RMSE when anchor variables were added. The average error difference favored the baseline models that relied only on the neighboring currency's history. Linear Regression produced similar outcomes, with little to no gain from including anchor features, suggesting that any linear effects were minimal or quite weak to influence our forecasts. KNN results were similar on individual pairs, and there were no systematic improvements attributed to anchor inputs. The dominant source of predictive signal came from currency's own lagged and moving average features rather than information from anchors.

PCA revealed that even the first two principal components capture meaningful separation between currency blocs, with EUR-linked currencies forming the most coherent group. K-Means clustering (k = 3) showed that EUR has the strongest follower alignment (80%), while CNY and JPY display weaker and most diffuse influence networks.

Related Work

Currency Exchange Rate Prediction – GitHub. This study uses ARIMA and recurrent neural networks (RNNs) to forecast exchange rates relative to the U.S. dollar, evaluating models using mean squared error over short historical windows. Their approach uses a single dominant anchor (USD) and does not analyze cross currency influence. We extend beyond USD based predictions by examining multiple anchor currencies (EUR, CNY, JPY) and uncovering follower relationships using supervised (XGBoost, LSTM, Linear Regression) and unsupervised methods (PCA and K-Means). Rather than predicting only price levels, we focus on anchor follower dynamics and cluster based co-movement patterns.

Forex – GitHub. This study developed a streamlet-based FX forecast tool that combines Alpha Vantage exchange rate data with macroeconomic indicators from FRED. The project uses feature engineering, Gradient Boosting Regressor, and exponential smoothing to generate single pair forecasts. This project models one currency; our project investigates system-wide multilateral influence networks. We use standardized log returns, PCA for dimensionality reduction, and K-Means clustering to discover emergent blocs rather than forecasting a single pair. Our supervised models quantify dependency on non-USD anchors.

Prediction of Foreign Currency Exchange Rate – ScienceDirect. This study introduces a hybrid architecture combining transformers and LSTM networks to forecast FX rates using deep sequential modeling. It focuses on predictive accuracy using feature rich time series inputs and emphasizes performance against traditional baselines. Our study integrates unsupervised learning to identify latent anchor follower groupings. Instead of using a single model to predict price movements, we combine PCA and clustering with supervised methods to quantify structural dependencies across currencies. Our framing shifts from pure forecasting to discovery and explanation of influence networks.

Data Source

Our analysis draws on multiple publicly available financial and macroeconomic datasets from institutional and API-based sources. The primary exchange rate data was obtained from the IMF Exchange Rate Dataset(https://data.imf.org/en/Data-Explorer?datasetUrn=IMF.STA:ER%284.0.1%29). The dataset was downloaded as CSV files and contains variables such as country and frequency. We retrieved the entire dataset, which covers both monthly and quarterly observations.

To incorporate macroeconomic context, we used the FRED Gross Domestic Product for World (NYGDPMKTPCDWLD) series from the Federal Reserve Economic Data platform (https://fred.stlouisfed.org/series/NYGDPMKTPCDWLD). This dataset was downloaded in CSV format and includes variables such as date and value. The full dataset was retrieved, and the time coverage spans monthly and quarterly periods.

We also used Real GDP (purchasing power parity) data from the CIA World Factbook (https://www.cia.gov/the-world-factbook/field/real-gdp-purchasing-power-parity/country-comparison/). This source provides downloadable CSV files containing name, value, date of information, region, and ranking. The entire dataset was retrieved, and it reflects data for the year 2024.

Additional trade data was incorporated from the IMF Trade Dataset (imf_trade_0) located at https://data.imf.org/en/Data-Explorer?datasetUrn=IMF.STA:IMTS(1.0.0). The dataset was downloaded in CSV format and includes variables such as country, counter country, and frequency. The full dataset was retrieved and covers both monthly and quarterly data.

High-frequency financial data on currency returns was gathered using the Yahoo Finance API (yfinance). Data was retrieved through JSON responses that were converted into Pandas DataFrames within Python. The key variables include daily close prices and ticker symbols, and the retrieval window covered the last several years at a daily frequency.

These combined datasets provided the foundation for constructing standardized FX log returns, macroeconomic context, and cross-country linkages necessary for supervised and unsupervised modeling.

Feature Engineering

To construct modeling ready features for both supervised and unsupervised learning, we developed a multi-stage pipeline beginning with raw macroeconomic and trade data sourced from IMF, FRED, and Yahoo Finance. This process includes extensive preprocessing, anchor neighbor mapping, removal of unsuitable currencies and feature generation based on time series returns.

We first cleaned and standardized the macroeconomic data. The FRED world GDP dataset contained the variables observation_date and NYGDPMKTPCDWLD. We converted the observation dates into datetime format, extracted the corresponding year to create a time_period column, and renamed the GDP column to world_gdp for clarity. This ensured consistency across all time related datasets.

The IMF GDP dataset was loaded using parsing parameters (quoting=csv.QUOTE_ALL, skipinitialspace=True) to avoid formatting issues. We standardized column names to lowercase with underscores and filtered observation by three criteria. First, the indicator measured “US Dollar per domestic currency”, second, the transformation type was “End-of-period”, and lastly, the data frequency must be annual. Year values were converted from strings to integers to enable temporal filtering.

For the trade data, we imported the IMF's trade dataset with the following fields: country, counterpart_country, time_period, obs_value, trade_flow, scale, and unit. A custom exclusion list was applied to remove aggregate regions. Columns were again standardized, and the time_period was cast to integer format. We retained only country_level export and import pairs with valid numeric trade values.

Anchor currencies were selected using the largest GDP economies outside the United States. Because the US dollar exercises dominant global influence rather than regional influence, including it as anchor would obscure the localized currency relationships we sought to analyze. We identify our three main anchor currencies as the following: Euro (EUR), Chinese Yuan (CNY) and Japanese Yen (JPY). To ensure the anchor neighbor relationship, we applied a few exclusion criteria. First, countries using the same currency as the anchor were removed, second, currencies pegged to the anchor were also excluded. This filtering step will ensure neighbors are independent.

We use IMF's trade data to link each potential neighbor to its respective anchor currency, we use the following calculation:

ExportShare(i, a) = Exports from country i to anchor a / Total exports of country i
ImportShare(i, a) = Imports from anchor a to country i / Total imports of country i

CombinedExposure(i, a) = (Exports from i to a + Imports from a to i) / (Total exports of i + Total imports of i)

We only retained countries as neighbors if their combined trade exposure to the anchor was at least 5%, ensuring each neighbor shows strong economic ties.

To further help us identify if movements of anchor currency can be used to help explain the future movements of neighbor currency, a Granger causality analysis test using daily log returns over the past 10 years was run on each pair of anchor and neighbor currency. Each pair was tested with lag lengths from 1 to 10 days using SSR F test set at 5% significant level. The lowest p-value is used to determine the best_lag for that pair, which represents the anchor currency number of days in its return history with the best predictive sign. Pairs without significant causality are then excluded from the supervised learning step.

Supervised Learning
To evaluate whether anchor currency information improves short-term FX return forecasting, we implemented three supervised learning models representing diverse methodological families: Linear Regression, K-Nearest Neighbors, and XGBoost. All models were trained under two experimental conditions: (1) Training_with_Anchor, where both neighbor and anchor features were included, and (2) Training_without_Anchor, where only the neighbor's own time-series features were used.

For each anchor neighbor currency pair, we constructed standardized daily log returns and engineered additional predictive features, including moving averages and lagged return values. We chose moving averages of 30, 60, and 90 as these are 1, 2, and 3 months respectively - assuming that a quarter of variation in returns would be enough to assess a trend. To add additional features we created daily lags ranging from 1 to 25 days as well - we felt this would identify intra-month trends. The target was defined as the one step ahead return of the neighboring currency. Linear Regression was selected as a transparent baseline to test whether linear dependencies from anchor currencies contribute additive predictive power. K-Nearest-Neighbors (KNN) was used to evaluate prediction based on local similarity in lagged and moving average space, offering an alternative not constrained

by linear assumptions. Lastly, XGBoost was chosen to model potential nonlinear interactions and cross series effects, using a gradient boosted trees to capture structure missed by simpler models.

For our linear regression model, no hyperparameter tuning was required beyond fitting on the chosen feature sets. For our KNN, it was evaluated with varying values of k to ensure stable neighborhood base predictions, though no consistent gains were observed from anchor inclusion. Lastly for our XGBoost, we used a grid search setup to optimize learning rate, depth, subsampling, and estimators under each variant. Linear Regression and XGBoost were evaluated using 5-fold cross-validation to ensure robustness and comparability across anchor and baseline settings. KNN was included for model diversity but evaluated without cross-validation due to runtime and performance considerations.

KNN regression was implemented as a nonparametric instance based learning model to analyze if currencies with similar recent movements (lags and average) would have shown similar future returns, which might not have been captured in other learning models. For each anchor neighbors pairs, neighbor currency was used 1 to 3 day lags and we used 1 to best_lag (up to 10) days for the anchor currency, based on Granger causality analysis of the pair. Features were standardized for a fair distance comparison. KNN regression was included due to the model's ability to learn from similar examples, which serves as a useful contrast to both the linear regression and XGBoost models.

We used Root Mean Squared Error (RMSE) as the primary evaluation metric, as FX returns are continuous and small magnitude difference matters in forecasting accuracy. Reporting RMSE enables comparison of overall predictive precision across families, while penalizing errors and returning in the same scale as training/test data units.

For each family, we selected the variance (anchor or without anchor features) that achieved the lowest cross-validated RMSE. As shown below, the baseline (no-anchor) setup consistently outperformed the anchor-augmented version.

Table 1.1

Model	Variant Used	Mean RMSE	Std Dev.
Linear	Training without anchor	0.000000000000009190932	4.169549E-15
XGBoost	Training without anchor	0.001556897	0.014967767
KNN	Training with anchor	0.00706	0.00361

Across all three models, including anchor features generally did not reduce forecasting error, KNN was the exception with improvement when anchor information was included. This suggested a limited short term influence of anchor currency movement that is detectable by instance based learning methods. Overall, Linear Regression’s baseline variant shows near zero error due to either feature leakage of complete linear dependence in the constructed dataset.

Among the three model families, XGBoost provided the most meaningful structure for ablation analysis. Removing anchor-based features and retraining showed no degradation in predictive accuracy, confirming that neighbor currency lags and moving averages dominate the signal. The tree-based splits further emphasized recent return dynamics over any cross-series input.

A sensitivity sweep across XGBoost hyperparameters (e.g., max_depth, learning_rate, n_estimators) showed that performance was relatively stable once anchor features were removed. In contrast, when anchor features were included, the model exhibited greater variance and less stable convergence, suggesting noise or weak cross-series dependencies.

The supervised models revealed several key tradeoffs across families. Linear Regression trained almost instantly and delivered near-perfect results in the baseline setting, but its sensitivity to feature leakage limits the reliability of those outcomes. In contrast, XGBoost required more computation but produced performance that was both stable and robust across currency pairs. The models also differed in their interpretability—Linear Regression offered transparent coefficients and a clear structure, whereas XGBoost captured nonlinear relationships at the cost of reduced explainability. Finally, expanding the feature space with anchor inputs increased complexity without improving accuracy, adding unnecessary computational overhead in both Linear Regression and XGBoost.

In-Depth Evaluation

For the purposes of our in-depth evaluation we focused on our Linear Regression model as it had the lowest RMSE of all three models. Each feature set resulted in ~30 unique features, we will share results of the top 5 in the interest of brevity.

- Ablation Analysis
    - Ranked in order of lowest to highest RMSE (lowest being most impactful) our top five features were:
        - MA_30: 7.484612e-15
        - return_lag_16: 7.790728e-15
        - return_lag_5: 8.296831e-15
        - MA_90: 8.374223e-15
        - return_lag_19: 8.441874e-15
    - Ranked in order of highest to lowest RMSE (highest being the least impactful) our top five features were:
        - return_lag_10: 1.460206e-14
        - return_lag_6: 1.133439e-14
        - return_lag_22: 1.106784e-14
        - return_lag_4: 1.082667e-14
        - return_lag_3: 1.061220e-14
- Feature Importance
    - Ranked in order of coefficients with highest correlation (highest being most impactful) our top five features were:
        - MA_60: 0.010453
        - return_lag_12: 0.000003
        - return_lag_3: 0.000003
        - return_lag_18: 0.000002
        - return_lag_11: 0.000001
    - Ranked in order of coefficients with lowest correlation (lowest being least impactful) our top five features were:
        - MA_30: -0.094279
        - MA_90: -0.090129

        - return_lag_15: -0.000004
        - return_lag_13: -0.000003
        - return_lag_22: -0.000003
- Sensitivity Analysis
    - Table 1.2
    -

```
Mean RMSE and standard deviation RMSE (in that order) - without hyperparameter tuning
 option
 Training_with_Anchor       6.801129e-03
 Training_without_Anchor    8.939379e-15
 Name: rmse, dtype: float64,
 option
 Training_with_Anchor       2.559958e-03
 Training_without_Anchor    5.607145e-15
 Name: rmse, dtype: float64)


Mean RMSE and standard deviation RMSE (in that order) - with hyperparameter tuning
option
Training_with_Anchor       6.846481e-03
Training_without_Anchor    2.488245e-18
Name: rmse, dtype: float64
option
Training_with_Anchor       2.738976e-03
Training_without_Anchor    1.206023e-18
Name: rmse, dtype: float64
```

    - Our linear regression model was sensitive to hyperparemeter tuning only in the case of training without anchors (.set_params(copy_X=False, fit_intercept=False, n_jobs=-1, positive = True)).
- Identifying Tradeoffs
    - Given the above results, evaluation and metrics there is a tradeoff to performing more moving average feature engineering, our top performing features tended to be moving averages. To improve local performance (we mostly had shared access and skillsets with small commodity hardware) we only engineered ~30 features. Done again I would suggest we increase the amount of training data, and generate a number of other moving averages - specifically MA_120, MA_150, MA_180 etc.

Failure Analysis

- First Example (target variable = ‘TND’)
    - Overview: Our linear regression model consistently predicted poorly for the target currency of ‘TND’ (see image below). Specifically it was -.061 off on the first prediction where the actual was .063 and the predicted was .002.
    - Table 1.3
    -

predictions	actuals	target	diff
0.002	0.063	TND	-0.061
0.000	0.060	TND	-0.060
-0.000	0.060	TND	-0.060
0.003	0.063	TND	-0.060
0.001	0.060	TND	-0.059

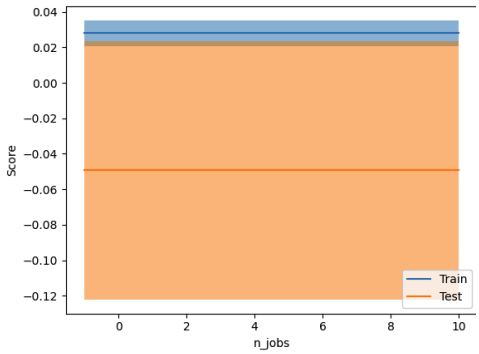
    - Category of failure: Systematic Errors
    - Cause: Data Related Issues

        - The training data for TND looks to have been underrepresented as either this country doesn't openly report on currency exchange rates or the data is flawed. The mean exchange rate return is '.000' which is directly related to the low prediction accuracy of our model.
    - Future Improvements: We could remove this from the training data set entirely.
- Second Example (target variable = CZK)
    - Overview: Our linear regression model was too simple of a model and not sensitive enough to hyperparameters, which caused the model to predict 'CZK' incorrectly, namely here - where the actual was -.007 and the predicted return was .0014 with a difference of .021.
    - Table 1.4
    -

predictions	actuals	target	diff
0.014	-0.007	CZK	0.021
0.001	-0.008	CZK	0.009
0.007	-0.001	CZK	0.008

    - Category of failure: Model Limitations
    - Cause: Data Related Issues
        - The linear regression model we built was too simple to be fine-tuned for our use case, it returned the same predictions no matter the number of n_jobs for example. This is a possible cause for why.
    - Future Improvements: We could choose a model that is more accessible to fine-tuning.

Image 1.1



- Third Example (target variable = SEK)
    - Overview: Our linear regression model would have performed better in certain scenarios where the anchor country feature was included. Specifically where the target was "SEK", note the diff decreased from .033 to .018 when the anchor was exposed in the training dataset.
    - Table 1.5
    -

predictions	actuals	target	option	diff
0.027	-0.006	SEK	Training_without_Anchor	0.033
0.017	-0.002	SEK	Training_without_Anchor	0.019
0.000	-0.018	SEK	Training_with_Anchor	0.018

    - Category of failure: Feature representation
    - Cause: Lack of features for purposes of study
        - We withheld anchor countries in aggregate or exposed them in aggregate
    - Future Improvements: Expose anchors to targets where there is an increase in model prediction accuracy to boost results.

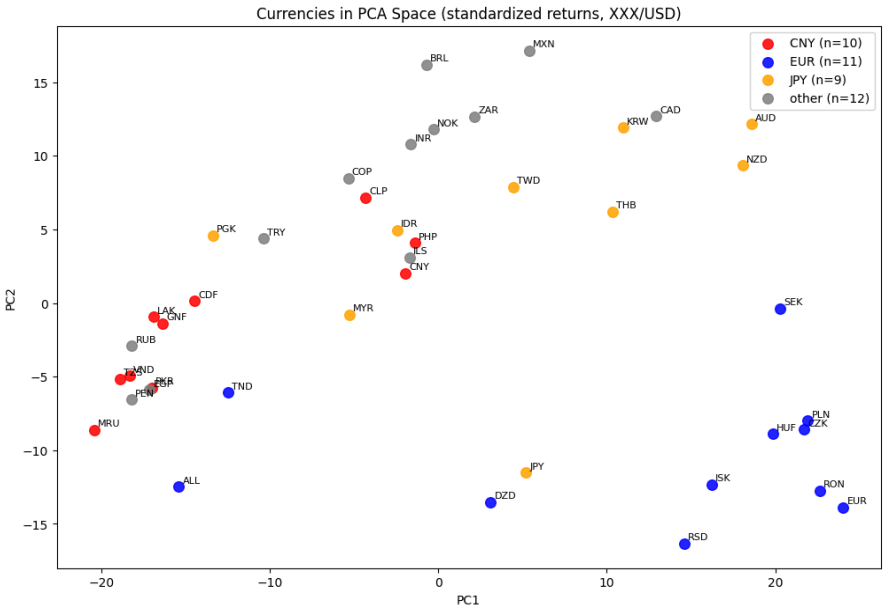
Unsupervised Learning

The unsupervised pipeline used daily FX closing rates for our anchor and neighbors' currencies at a one-day frequency obtained through Yahoo Finance symbols. We converted all series to XXX/USD levels and inverted when only USD/XXX was available. Daily log returns were computed for each

currency and standardized to remove scale and trend effects, ensuring the methods focused on movement structure rather than nominal levels. Short gaps of up to three days were forward filled; any rows with remaining missing values were dropped. Standardized log returns were chosen as the feature representation because they emphasize co-movement behavior and allow consistent comparison across currencies.

PCA (non-probabilistic, linear) was used to identify orthogonal directions that explain the greatest share of variance in the time–currency matrix. We fit PCA on the transposed feature matrix and retained the top two principal components (n_components = 2) for visualization and downstream clustering. The loadings were interpreted to understand anchor–neighbor structure. PCA was chosen as a dimensionality reduction method that exposes latent factors and provides an interpretable basis for analysis and denoising. We considered using more components but selected two because the first

Image 1.2



two principal components captured the majority of shared variance and approximated the elbow in the explained variance curve. Projecting currencies into the first two principal components reveals clear

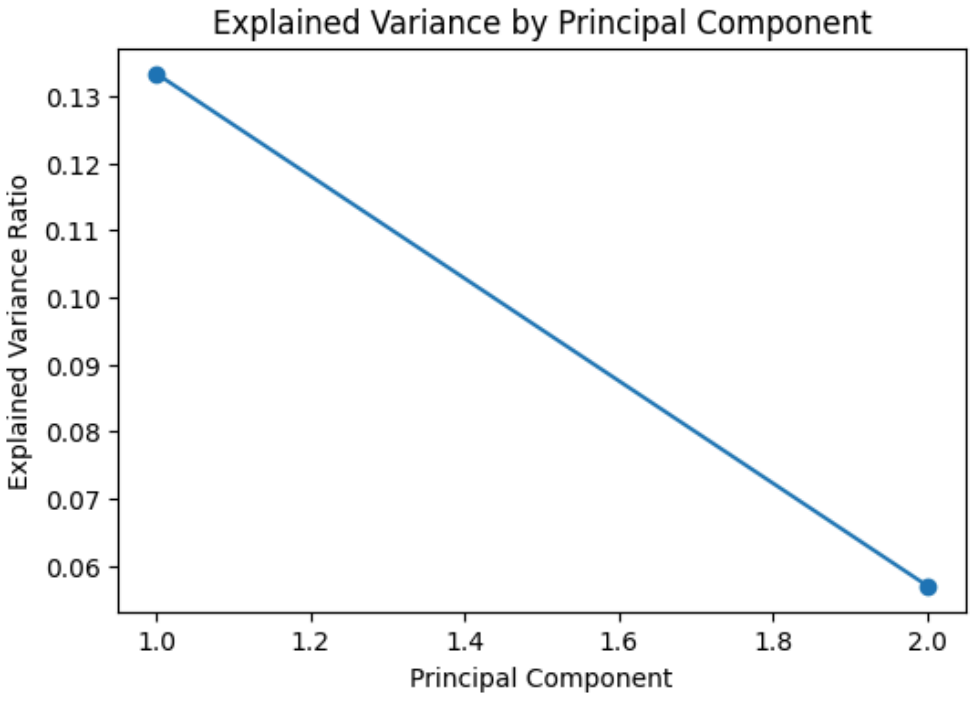


Image 1.3

spatial separation with economic groupings. EUR linked currencies cluster on the far right, CNY and JPY affiliated currencies appear more dispersed toward the left and center, showcasing distinct co-movement structures.

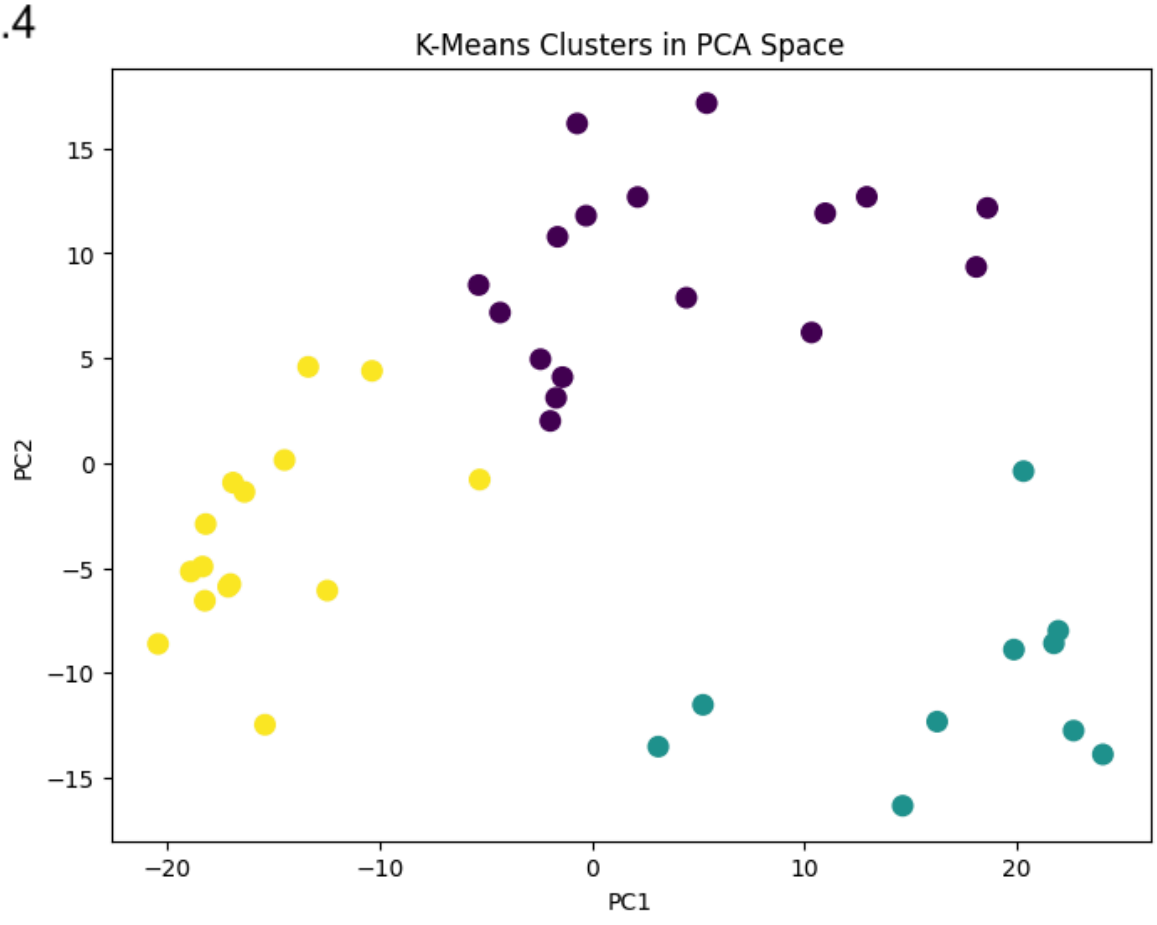
The first principal component explains roughly 13% of total variance, and the second adds another 6%. The increase drop after PC2 indicates an elbow, confirming that retaining only two components is sufficient to capture the strongest common structure among currencies for clustering.

K-Means (instance-based, centroid model) was applied to the PCA-transformed currency coordinates to partition currencies into groups by minimizing within-cluster Euclidean distance. We used k = 3 and random_state = 42 for consistency. K-Means was selected because it provides simple, transparent cluster centers and is well suited to convex structures in PCA space. The choice of k = 3 was based on both interpretability and the expectation of anchor-based grouping behavior.

Clustering on the 2D PCA projection, three clearly separated groups, confirming that k = 3 identifies meaningful structure rather than noise. The spatial separation reflects distinct co-movement behavior among the currencies.

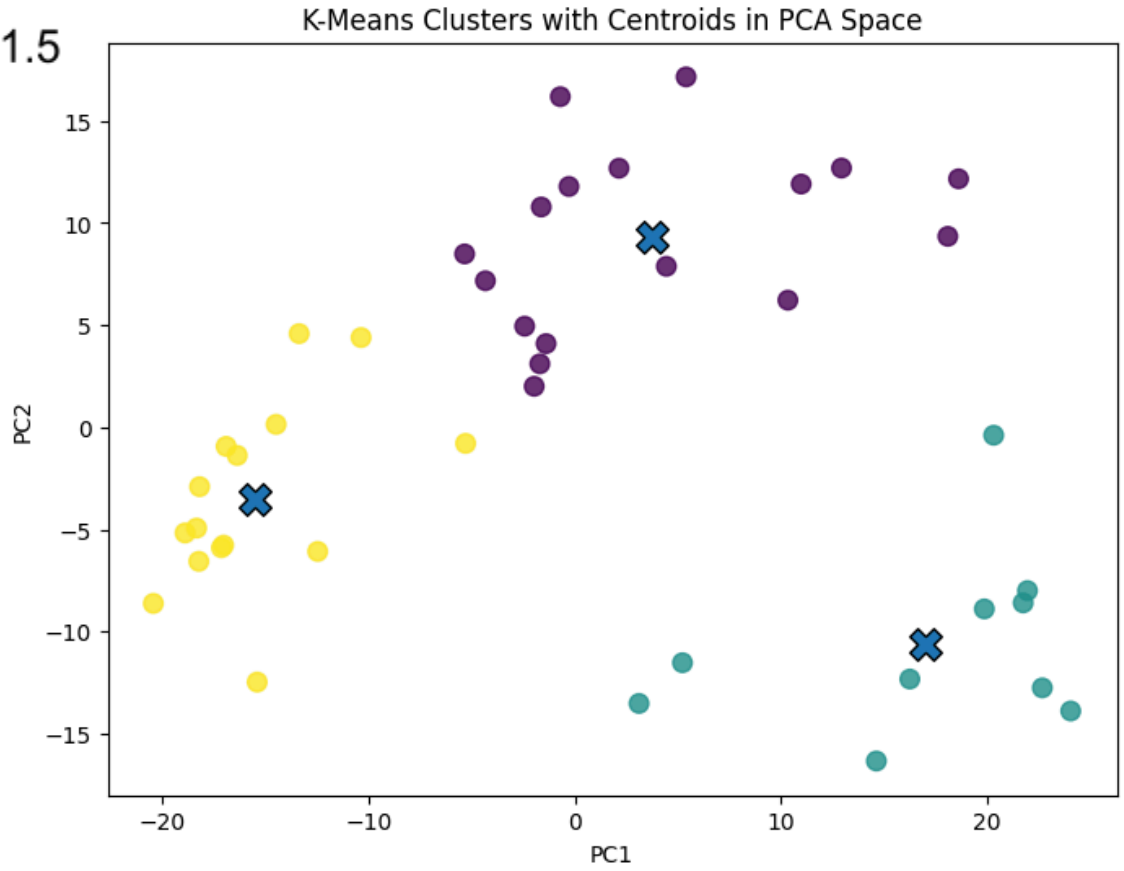
Adding cluster centroids shows that each group is compact and well-centered, further supporting k=3 as an appropriate choice. The centroid positions reinforce the stability and cohesive of each cluster in reduced space.

Image 1.4



We used different forms of evaluation to assess structure, coherence, and anchor relevance in the clusters. For PCA, we examined the explained variance ratio. PC1 accounted for 13.34% and PC2 for 5.59% of the variance, with a cumulative total of 19.02%, which is sufficient to reveal anchor-driven movement patterns. For K-Means, we used the silhouette score to assess cohesion and separation; a

Image 1.5



score of 0.55 indicated meaningful cluster structure in PCA space. We also computed an external anchor match metric to measure how many of each anchor's known neighbors were assigned to the same cluster: EUR showed 80% alignment (p = 0.001), CNY showed 22.2% alignment (p = 0.863), and JPY showed 0% alignment (p = 1.0). These findings suggest that EUR forms the most coherent and directionally influential bloc, while CNY and JPY exhibit more dispersed cluster membership.

- Table 1.6

Method	Configuration	Key Metrics
PCA	n_components=2	PC1 = .1333374 PC2 = 0.56873 Cumulative = 0.19024773
K-Means	k=3, random_state=42	Silhouette = 0.55200253 Anchor_Match: EUR = 0.8 CNY = 0.22222 JPY= 0.0

Discussion

From Part A, we learned how different supervised learning models behave to the inclusion of anchor currency data in FX forecasting. We learned that anchor based predictive features did not provide consistent benefits across models. Despite testing three fundamentally different supervised methods, Linear Regression, KNN, and XGBoost, the dominant predictive signal came from each currency's own historical returns rather than cross-series anchor inputs. This indicates short horizon FX predictability is primarily self driven; any influence from anchor currencies may not be meaningful at the daily lag structure we modeled.

A surprise outcome was the complete absence of performance gains in XGBoost when anchor features were included. We originally expected nonlinear methods to extract at least some small improvements, especially for clusters like EUR, but the RMSE differences favored baseline models without anchor inputs. However, KNN, which relies on similarity in feature space, shows improvements demonstrating that anchor influence is not significant at short horizons. Statistical significance test (paired t-test at $p<0.1$) was run on our result of the model's performance. KNN model, while showing improvement, is not statistically significant, meaning the RMSE between the model with and without anchor features was likely because of chances. For the linear regression model and XGBoost model, the result was shown that adding anchors statistically significantly reduces the model prediction accuracy.

One of the main challenges in Part A was constructing consistent with/without anchor feature representation across all models. Ensuring no data leakage and keeping features aligned across models required repeated restructuring of the pipeline. We address by standardizing lag generation, automating anchor exclusion, and using RMSE under time aware validation.

With more time and resources, we would extend this experiment with restructuring original data, more lag windows, alternative targets. Though we explore Granger Causality briefly, a deeper and more detailed analysis into longer lag relationships could potentially provide further insights.

From Part B, we analyzed unsupervised patterns in the return of FX to identify regional structural and common movements of currency returns. We learned that meaningful latent structure exists in standardized FX log returns, even when only a small number of principal components are retained. PCA revealed that a limited set of factors governs moment across currencies, and K-Means clustering on the reduced space exposes distinct blocs with differing coherence levels. This validated our assumption that anchor follower dynamics can be discovered.

The most surprising finding was the strength of the EUR center grouping. While JPY and CNY exhibited weaker networks, EUR achieved an 80% alignment in its cluster, suggesting stronger follower behavior in practice than anticipated. This contracted with our expectation that CNY would produce a similar bloc due to its regional and economical influences.
We encountered challenges related to uneven data availability across currencies. Some series required forward filling, inversion, or removal due to missing values, and we need to ensure that preprocessing did not distort cluster structure. Running PCA on standardized returns and using only currencies with sufficient history helped address this.

If given more time, we would like to test alternative clustering techniques, for example maybe hierarchical clustering. We may also look into dynamic clustering over rolling windows to study time evolving anchor influence.

Ethical Considerations

Part A - Supervised Learning
Using supervised models to predict or classify currency relationships or macroeconomic behavior introduces several ethical risks. One key concern is misinterpretation and overconfidence: supervised predictions could be treated as deterministic signals for financial decisions, leading to unintended consequences for markets, investors, or policymakers. To mitigate this, we provide clear communication on model uncertainty, avoid prescriptive claims, and frame predictions as exploratory or probabilistic rather than definitive.

A second issue involves data quality and bias. If training data reflect unstable economic events, outliers, or uneven country representation, the model may propagate biased assumptions that disadvantage smaller or emerging economies. We address this risk by evaluating input data coverage across regions, using diagnostics to detect bias, and reporting limitations of model generalization.

There is also the ethical risk of misuse for speculation or exploitation. Forecasts could be applied irresponsibly by institutions or governments to manipulate markets or influence currency-dependent economies. To mitigate this, we emphasize responsible use, restrict claims about real-world deployment, and maintain transparency about the intended academic scope of the project.

Part B - Unsupervised Learning
Unsupervised methods like PCA and K-Means introduce different ethical risks, particularly around interpretation. Overinterpretation of clusters is a concern, as cluster groupings may be incorrectly assumed to reflect geopolitical, economic, or cultural relationships, leading to oversimplified narratives about regions or currencies. To mitigate this, we emphasize that clusters are statistical groupings based on returns rather than intrinsic economic or political identities.

There is also risk from data exclusion and representation. Forward filling gaps and removing currencies with insufficient data may bias structure toward countries with more reliable financial reporting. To address this, we report exclusion criteria, acknowledge missingness, and note that “other” clusters may reflect data artifacts rather than meaningful behavior.

Statement of Work

Joey contributed primarily to the supervised learning components of the project. He developed most of the supervised model code, handled data engineering tasks, set up both the Deepnote environment and the GitHub repository, and participated in writing sections of the paper. Phuc focused on extending and refining both the supervised and unsupervised learning pipeline as well as feature engineering for anchor neighbor pairs, implementing Granger causality for pairs filtering and lag selection. Building on the initial framework, he worked with the PCA and K-Means components, KNN modeling, and contributed to writing the report. Spencer initiated the unsupervised learning codebase and assisted in the data acquisition and cleaning process, including testing and pulling FX data through yfinance. He also generated visualizations and contributed to writing the paper.

References

You must include a reference section at the end of your report. Use standard citation style to include references to any papers, projects, web sites, or other resources you used for your project. References do not count toward the page limit.

- Kushg18. (2018). *Currency exchange rate prediction* [Source code]. GitHub. https://github.com/kushg18/currency-exchange-rate-prediction
- Raulincadet. (2021). *Forex: A machine learning dashboard for predicting exchange rates* [Source code]. GitHub. https://github.com/raulincadet/forex
- Ghahremani, S., & Nguyen, U. T. (2025). Prediction of foreign currency exchange rates using an attention-based long short-term memory network. *Machine Learning with Applications, 20*, 106468. https://doi.org/10.1016/j.mlwa.2025.106468
- Colab Notebook – LSTM Time Series. Macedo, D. (n.d.). *Time Series Prediction with LSTM using PyTorch*. Google Colab. https://colab.research.google.com/github/dlmacedo/starter-academic/blob/master/content/courses/deeplearning/notebooks/pytorch/Time_Series_Prediction_with_LSTM_Using_PyTorch.ipynb
- XGBoost with GridSearchCV (Kaggle). Jayatou. (n.d.). *XGBRegressor with GridSearchCV*. Kaggle. https://www.kaggle.com/code/jayatou/xgbregressor-with-gridsearchcv
- RMSE Cross-Validation in sklearn (StackOverflow). StackOverflow. (2021). *RMSE cross validation using sklearn*. https://stackoverflow.com/questions/69432869/rmse-cross-validation-using-sklearn
- Configuring XGBoost via Dict (XGBoosting.com). XGBoosting. (n.d.). *Configure XGBoost model with parameters defined in a dict*. https://xgboosting.com/configure-xgboost-model-with-parameters-defined-in-a-dict/
- RMSE Loss in PyTorch (PyTorch Forum). PyTorch Discuss. (2017). *RMSE loss function*. https://discuss.pytorch.org/t/rmse-loss-function/16540
- Granger Causality Test in Python (Statology). Statology. (2022). *Granger Causality Test in Python*. https://www.statology.org/granger-causality-test-in-python/
- KNN Classification Overview (GeeksforGeeks). GeeksforGeeks. (n.d.). *K-Nearest Neighbours (KNN)*. https://www.geeksforgeeks.org/machine-learning/k-nearest-neighbours/
- KNN Regression in scikit-learn (GeeksforGeeks). GeeksforGeeks. (n.d.). *K-Nearest Neighbors (KNN) Regression with scikit-learn*. https://www.geeksforgeeks.org/machine-learning/k-nearest-neighbors-knn-regression-with-scikit-learn/
- Cross-Validation in PyTorch (Medium). Biased Algorithms. (n.d.). *Cross-validation in PyTorch*. Medium. https://medium.com/biased-algorithms/cross-validation-in-pytorch-2f9f9fa9ab16

Appendix

- GitHub: https://github.com/joehiggi/siads-696.git
  - Direct Link to Compiled Jupyter Notebook: https://github.com/joehiggi/siads-696/blob/main/src/siads_696_1.ipynb
- Data: https://drive.google.com/drive/folders/1yxh53sPbtPPKaZ3giQFF8WvMb0Ky-3v2?usp=share_link