

Machine Learning Methods for FDI Growth Forecasting

Introduction to Advanced ML Techniques

Patricio¹

¹Department of Economics, Stanford University

¹Department of Statistics, Stanford University

Winter 2025

Overview

- 1 Introduction to Methods
- 2 Principal Component Analysis (PCA)
- 3 Purged k-fold Cross-Validation
- 4 Bagging (Bootstrap Aggregation)
- 5 Random Forests
- 6 Summary

Overview of Advanced ML Techniques

Our 4-Step Sequential Implementation:

- ➊ **PCA Transformation:** Pre-process features to address multicollinearity
- ➋ **Purged k-fold CV:** Prevent information leakage in time series
- ➌ **Bagging (Bootstrap Aggregation):** Reduce forecast variance
- ➍ **Random Forest:** Add feature randomization for robustness

Benefits for FDI Forecasting:

- Captures **non-linear dependencies** between economic indicators
- Systematically reduces **forecast variance and bias**
- Identifies **key FDI drivers** through feature importance analysis
- Prevents **overfitting** to historical patterns

Principal Component Analysis (PCA)

Mathematical Formulation:

$$\mathbf{Z}_{t,n} = \sigma_n^{-1}(\mathbf{X}_{t,n} - \mu_n) \quad (\text{Standardize}) \quad (1)$$

$$\mathbf{Z}'\mathbf{Z}\mathbf{W} = \mathbf{W}\mathbf{\Lambda} \quad (\text{Eigendecomposition}) \quad (2)$$

$$\mathbf{P} = \mathbf{Z}\mathbf{W} \quad (\text{Project onto principal components}) \quad (3)$$

Benefits for FDI Forecasting:

- Decorrelates features while preserving information
- Reduces dimensionality to 95% explained variance
- Filters measurement noise in economic indicators
- Speeds computation by reducing tree depth requirements

Why? Economic indicators driving FDI show significant collinearity; PCA reveals the fundamental drivers while improving model stability.

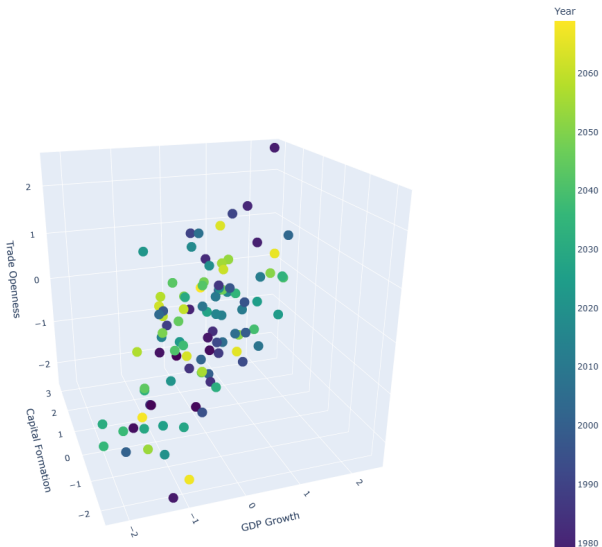
Principal Component Analysis (PCA)

PCA Steps:

- 1 Standardize variables to unit variance
- 2 Compute covariance/correlation matrix
- 3 Find eigenvectors and eigenvalues
- 4 Project data onto principal components

PCA Visualization: 3D Example

Original Economic Indicators Data (3D)



PCA Decomposition and Projection

Key Insights:

- First principal component captures maximum variance
- Each subsequent component is orthogonal to previous ones
- For our FDI analysis, we retain components that explain 95% of variance

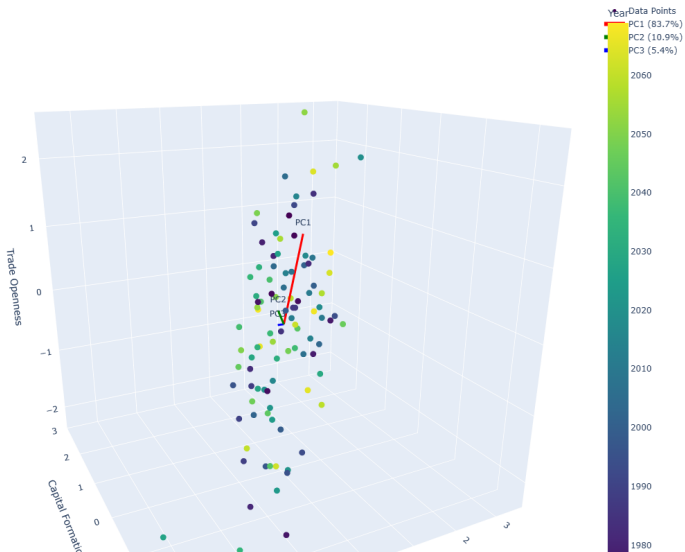
PCA Applied to FDI Drivers I

FDI Application:

- PC1 represents **Capital Development** (capital stock, effective exchange rate)
- PC2 primarily captures **Market Conditions** (population growth, labor metrics)
- PC3 relates to **Inflation and Monetary Policy**
- PC4 primarily captures **Trade Openness**

PCA Applied to FDI Drivers II

Economic Indicators with Principal Components



Why Financial Time Series Need Special CV

Problems with Standard CV for Time Series:

- **Information leakage** between training and testing
- **Label overlaps** lead to overly optimistic performance estimates
- **Temporal dependencies** aren't respected with random sampling
- **Autocorrelation** between nearby observations

Solution: Purged k-fold Cross-Validation with Embargo

- Maintains chronological order of observations
- **Purging**: Removes training observations whose labels overlap with test set
- **Embargo**: Adds buffer period after test set to prevent leakage
- Developed by Lopez de Prado in "Advances in Financial Machine Learning"

Purged k-fold CV Implementation

Mathematical Formulation:

- Train observation at t_i is purged if $[t_{i,0}, t_{i,1}] \cap [t_{j,0}, t_{j,1}] \neq \emptyset$ for any test observation j
- Embargo excludes training observations where $t_{i,0} \in [t_{j,1}, t_{j,1} + h]$



Figure: Purged k-fold Cross-Validation on Singapore FDI Data

Purged k-fold CV: Implementation Details

Configuration:

- Configure $k=5$, e.g., folds with preserved time ordering. We did 8.
- **Purging**: Remove training obs with labels overlapping test periods
- **Embargo**: Add buffer period ($0.01T$) after test set

Purged k-fold CV: Algorithm

Algorithm Purged k-fold Cross-Validation

```

1: Input: Data  $X$ , labels  $y$ , time intervals  $[t_0, t_1]$ , n_splits  $k$ , embargo pct  $e$ 
2: Split data chronologically into  $k$  test folds
3: for each test fold do
4:    $train\_indices \leftarrow$  all indices except current test fold
5:   for each test sample  $j$  with time interval  $[t_{20}, t_{21}]$  do
6:     for each potential train sample  $i$  with time interval  $[t_{10}, t_{11}]$  do
7:       if  $[t_{10}, t_{11}] \cap [t_{20}, t_{21}] \neq \emptyset$  then
8:         Remove  $i$  from  $train\_indices$  ▷ Purging
9:       end if
10:    end for
11:  end for
12:   $embargo\_size \leftarrow \text{round}(e \times \text{len}(X))$ 
13:  Remove training samples in  $[t_{20}, t_{21} + h]$  where  $h = 0.01T$  ▷ Embargo
14:  yield  $train\_indices, test\_indices$ 
15: end for

```

Bootstrap Aggregation (Bagging)

The Problem: High variance in financial time series forecasts due to:

- Limited historical data for FDI
- Structural changes in Singapore's economy
- Sensitivity to macroeconomic shocks

Bagging Solution:

- Reduce forecast variance by averaging multiple models
- Each model trained on different bootstrap samples
- Bootstrap samples drawn with replacement
- Variance reduction proportional to model decorrelation

$$V \left[\frac{1}{N} \sum_{i=1}^N \varphi_i[c] \right] = \bar{\sigma}^2 \left(\bar{\rho} + \frac{1 - \bar{\rho}}{N} \right) \quad (4)$$

where $\bar{\rho}$ is average correlation between predictions

Bagging Implementation for FDI Forecasting

Implementation:

- Generate $N = 1000$ bootstrap samples with replacement
- Set `max_samples` to average uniqueness between observations
- Train independent estimators on each resampled dataset
- Average predictions: $\hat{y}_{t+1} = \frac{1}{N} \sum_{i=1}^N f_i(x_t)$

Sequential Bootstrapping Extension:

- Account for temporal structure in FDI time series
- After selecting observation i , probability of selecting nearby observations reduces
- Probability decay: $p_j = p_j \times \text{decay}^{1/|i-j|}$
- Creates more diverse models for averaging

Bagging Picture

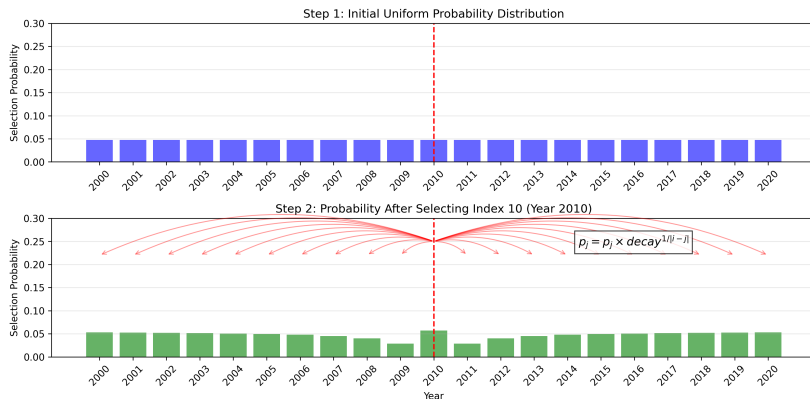


Figure: Sequential Bootstrapping with Probability Decay

Bagging Algorithm

Algorithm Sequential Bagging for FDI Forecasting

- 1: **Input:** Training data (X, y) , $n_estimators = 1000$, $decay_factor = 0.6$
- 2: Compute $avg_uniqueness = \text{uniqueness between observations}$
- 3: **for** $i = 1$ to $n_estimators$ **do**
- 4: Initialize uniform probabilities $p = [1/n, \dots, 1/n]$
- 5: $sample = []$
- 6: **for** $j = 1$ to $max_samples$ **do**
- 7: Select observation idx with probability p
- 8: $sample.append(idx)$
- 9: **for** each time point k **do**
- 10: $p[k] = p[k] \times decay_factor^{1/|k-idx|}$ ▷ Reduce probability
- 11: **end for**
- 12: Renormalize p to sum to 1
- 13: **end for**
- 14: Train decision tree f_i on bootstrap sample $(X[sample], y[sample])$
- 15: **end for**
- 16: Final prediction: $\hat{y} = \frac{1}{n_estimators} \sum_{i=1}^{n_estimators} f_i(X)$

Random Forests: Beyond Bagging

From Bagging to Random Forests:

- Bagging: Same features considered at each split
- Random Forests: **Random subset of features** at each split
- For regression tasks (like FDI): Use **$p/3$ features** at each split
- Further reduces correlation between trees

Configuration for FDI Forecasting:

- Base estimator: Decision trees with **MSE criterion**
- **$\text{max_features} = p/3$** for regression tasks
- **$\text{min_weight_fraction_leaf} = 0.05$** for regularization
- Sequential bootstrapping to reduce $\bar{\rho}$ between trees

Decision Rules for FDI:

- Each node represents a split on a feature (e.g., capital stock $>$ threshold)
- Leaf nodes contain predicted FDI values
- Tree depth limited to prevent overfitting
- Min samples per leaf to ensure generalization

Simple Decision Tree Example

Single Decision Tree for FDI Prediction

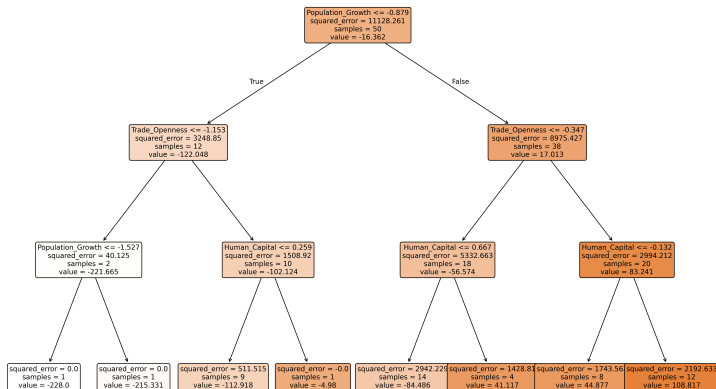


Figure: Single Decision Tree Structure

Random Forest (4 Trees)

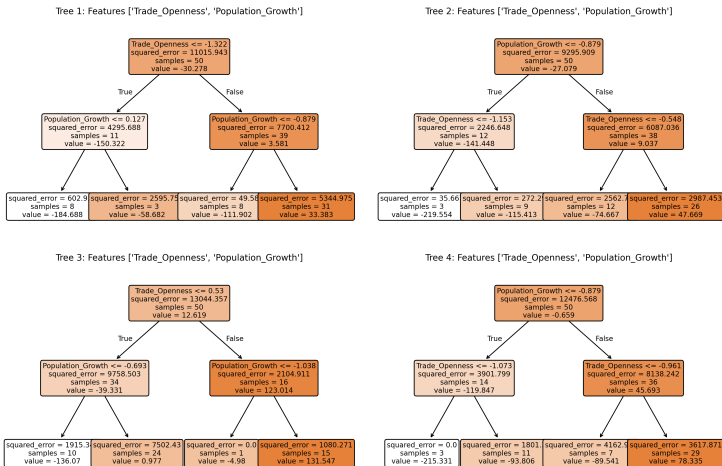


Figure: Multiple Trees Combined in Random Forest

Random Forest Ensemble

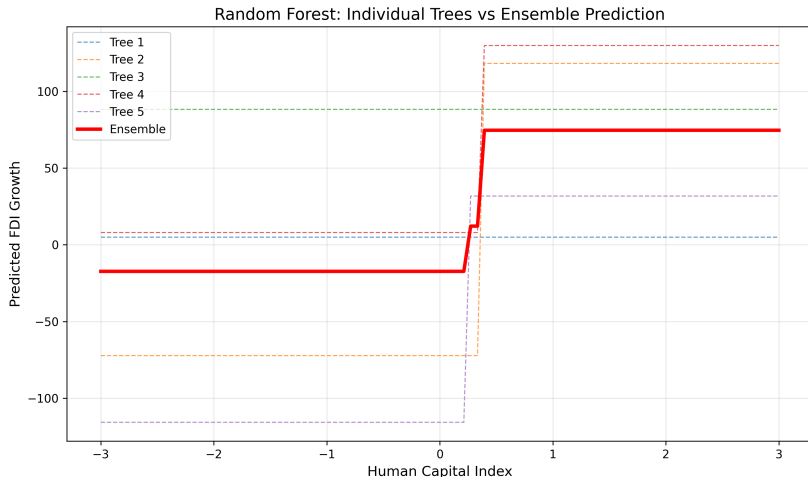


Figure: Trees with Different Predictions Combined in Random Forest

Feature Selection from ML Perspective - Final

Random Forest Feature Importance:

- Using Single Feature Importance (SFI): Evaluates **predictive power** of individual features
- Reverse PCA transformation to identify original feature importance
- Distribute PCA vector importance back to original features
- Weight by loading magnitude in principal components
- Select top 3 features for model comparison

Top 3 Features:

- 1 **Human Capital Index** (PC1): Represents capital development
- 2 **Population Growth** (PC2): Captures market size dynamics
- 3 **Trade Openness** (PC4): Measures integration with global economy

Result: Selected variables follow the hypothesized causal chain: Human Capital → Population Growth → Trade Openness → FDI Growth

Summary of ML Approach

Complete 4-Step Pipeline:

- ① **PCA Transformation:** Reduced dimensionality while preserving 95% variance
- ② **Purged k-fold CV:** Prevented information leakage with purging and embargo
- ③ **Bagging:** Reduced forecast variance through sequential bootstrapping
- ④ **Random Forest:** Added feature randomization for robust predictions

Key Advantages:

- Captures complex non-linear relationships in FDI determinants
- Provides feature importance for economic interpretation
- Minimizes overfitting through regularization and ensemble methods
- Identifies crucial variables for SVAR model comparison

Next Step: Compare against structured SVAR model using selected features

References

- [1] López de Prado, M. (2018). *Advances in Financial Machine Learning*. Wiley.
- [2] Esclanda-Lo (2024). Machine Learning for FDI Determinants in Developing Asian Economies.
- [3] Bhujabal (2024). Institutional Quality and Inward FDI in Southeast Asia.
- [4] Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32.