Predicting Economic Recessions Using Multi-Country Indicators

CAP4770 Data Mining - Final Project Report

December 2025

1. Problem Statement and Background

Recessions are among the most consequential economic events, affecting employment, investment, and quality of life for millions of people. Despite their significance, recessions remain notoriously difficult to predict. Traditional economic indicators often lag behind actual conditions, meaning a recession is typically confirmed only after it has already begun.

This project tackles the challenge of building a **forward-looking recession prediction system** using machine learning classification techniques. Rather than simply detecting whether we are currently in a recession, the goal is to provide early warning signals that a recession may be approaching in the coming months.

Problem Definition

The core question is: *Can we predict U.S. recessions before they occur using a combination of domestic and international economic indicators?*

This is framed as a binary classification problem where:

- Class 0 = No recession (economic expansion)
- Class 1 = Recession period

Why This Matters

Even a few months of advance warning could allow policymakers, investors, and businesses to prepare. However, there is an inherent trade-off: a model that catches every recession but generates many false alarms may cause unnecessary panic, while a model that minimizes false alarms might miss critical warning signs. This project explores that balance.

Design Decision: Excluding CPI/Inflation Data

A deliberate choice was made to exclude Consumer Price Index (CPI) and other inflation metrics from the feature set. While inflation data is economically relevant, it risks introducing bias because the Federal Reserve's response to inflation (raising interest rates) is itself a known recession trigger. Including CPI could lead the model to learn a confounded relationship rather than genuine predictive patterns.

2. Data Storage and Database Integration

All collected data was loaded into a MySQL database before being fetched into the Jupyter notebook environment for preprocessing, analysis, and modeling. This

approach ensures data integrity, enables efficient querying, and provides a reproducible pipeline from raw data to final analysis.

The database schema included tables for:

- US economic indicators (GDP growth, treasury rates, trade data)
- International indicators (EU, Japan, China, Korea, UK)
- Official recession date ranges from NBER
- Derived features (yield spreads, lagged variables)

Data was fetched from the database using pandas' *read_sql()* function, allowing seamless integration between the database backend and Python analysis environment.

3. Data Collection and Preprocessing

Data Sources

Economic data was collected from two primary sources. The Federal Reserve Economic Data (FRED) database, maintained by the Federal Reserve Bank of St. Louis, served as the main source for U.S. indicators including GDP growth rates, treasury yields, and official recession dates. The World Bank's Global Economic Monitor provided complementary international data for cross-border analysis.

Countries and Indicators

The analysis covers six major economies, chosen for their global economic influence and data availability:

- United States (primary focus)
- European Union
- Japan
- China
- South Korea
- United Kingdom

For each country, the following indicators were collected where available:

- GDP growth rate (quarterly, converted to monthly)
- 3-month treasury/government bond yield
- 10-year treasury/government bond yield
- Exports and imports (trade data)
- Official recession indicator (US only, from NBER)

Preprocessing Steps

The raw data required several preprocessing transformations:

1. **Temporal alignment:** All data was aligned to monthly frequency. Quarterly GDP data was forward-filled to monthly observations.
2. **Missing value handling:** Missing values were handled through forward-fill for time series continuity, with remaining gaps dropped.
3. **Feature engineering:** Yield curve spreads were calculated (10Y minus 3M rates), and trade balances computed (exports minus imports).

4. **Lagged features:** To enable forward-looking prediction, lagged versions of key features were created at 3, 6, and 12-month intervals.

The analysis period spans from 1990 to 2025, capturing three major U.S. recessions: the 2001 dot-com bust, the 2008-09 financial crisis, and the 2020 COVID-19 recession. While earlier data from the 1960s was available for some series, the 1990 cutoff was chosen to ensure consistent multi-country coverage.

4. Exploratory Data Analysis

Before building predictive models, the data was explored through visualizations to understand patterns and relationships. Three key visualizations reveal the structure of the prediction problem.

Dataset Overview

Metric	Value
Training Observations	297
Test Observations	119
Training Recession Months	26
Test Recession Months	2
Class Imbalance Ratio	~14:1 (non-recession to recession)

Table 1: Train/Test Split Summary

The severe class imbalance is immediately apparent. Recessions are rare events—only 28 months out of 416 total observations (6.7%) are classified as recession periods. This imbalance is actually representative of reality; recessions are infrequent but high-impact events. However, it creates challenges for model training, as classifiers can achieve high accuracy simply by predicting "no recession" for every observation.

Visualization 1: Logistic Regression Model with Economic Indicators

The first visualization presents a three-panel view of the Logistic Regression model alongside key economic indicators. The top panel shows the model's predicted recession probability over time, with pink shading marking actual recession periods. The middle panel displays the US Yield Curve Spread (10-year minus 3-month treasury rates), where red shading indicates periods of inversion. The bottom panel shows quarter-over-quarter GDP growth.

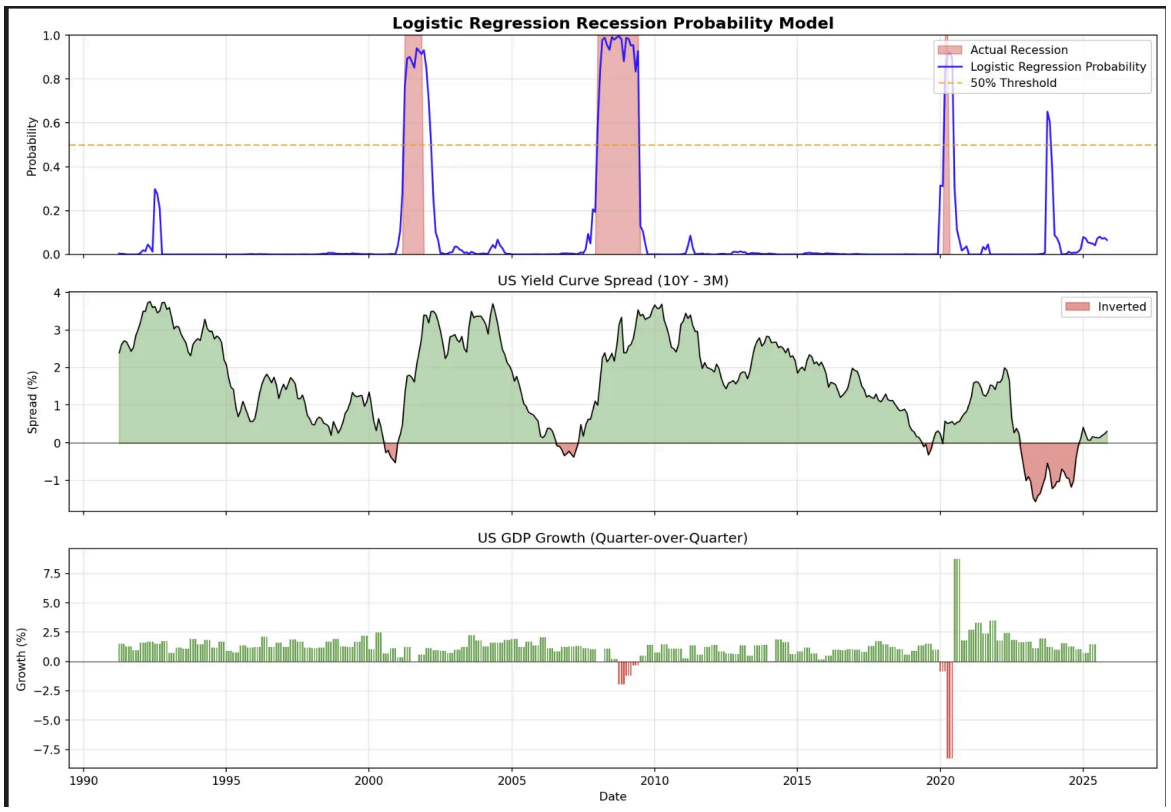

Figure 1: Logistic Regression Recession Probability Model

The yield curve inversions (red areas in middle panel) clearly precede each recession by several months, confirming its status as a leading indicator. The 2022-2023 inversion was the deepest on record at nearly -1.5%, which explains the elevated probability signals in recent years despite no recession materializing. The logistic regression model produces smooth, gradual probability changes and successfully spikes during all three historical recessions.

Visualization 2: K-Nearest Neighbors Model

The second visualization shows the same three-panel structure for the KNN model. The contrast with logistic regression is striking—KNN produces much sharper, more discrete probability signals.

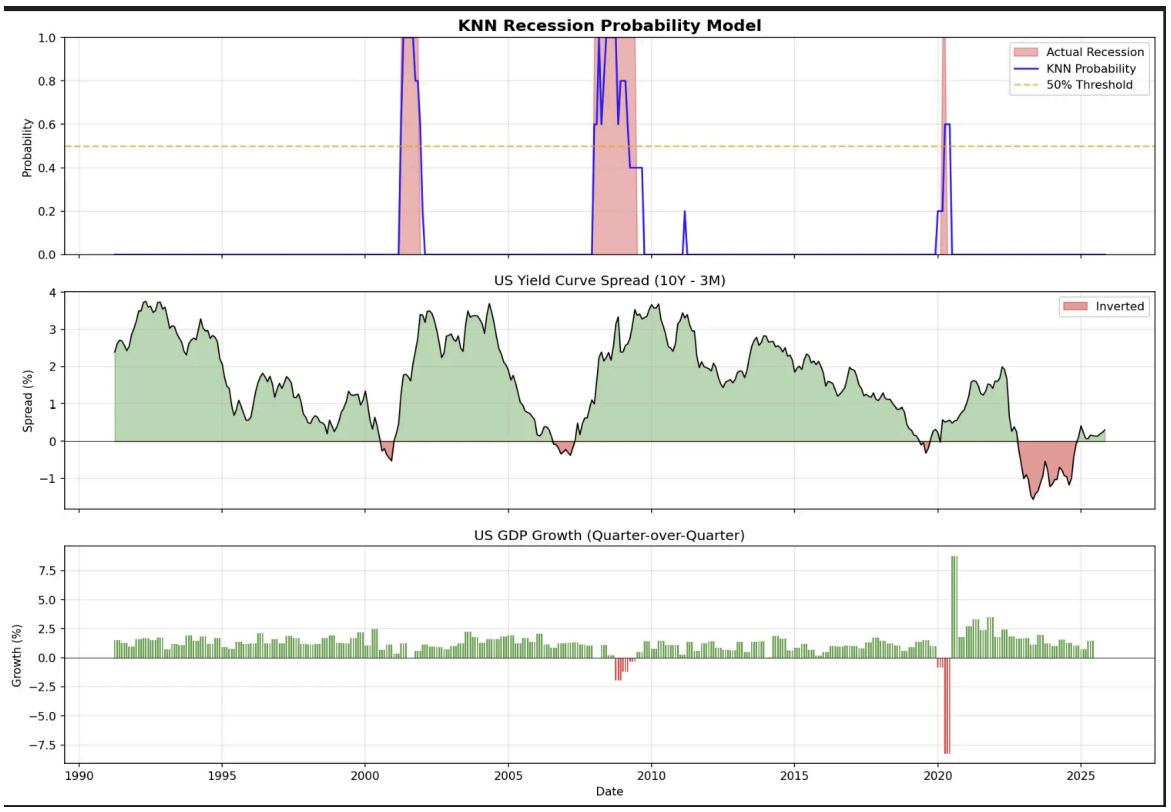

Figure 2: K-Nearest Neighbors Recession Probability Model

KNN's probability stays near zero during stable economic periods and rises sharply only when current conditions closely resemble historical recession patterns. The early 1990s noise visible in logistic regression is absent here. This cleaner signal explains KNN's superior precision—it only fires when conditions closely match past recessions, resulting in fewer false alarms while still detecting both test-set recession months.

Visualization 3: Random Forest Model

The third visualization reveals why Random Forest performed worst among the three models.

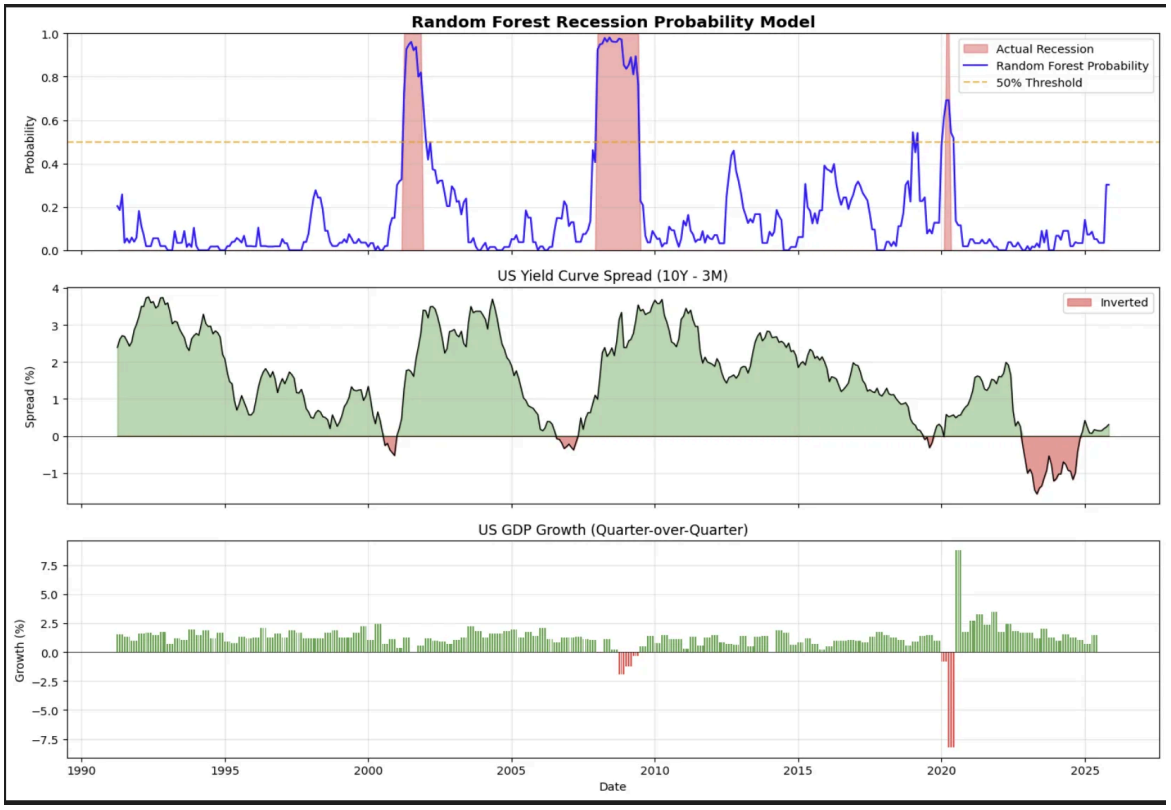

Figure 3: Random Forest Recession Probability Model

The Random Forest probability signal is noisy and erratic, fluctuating between 0.1-0.5 even during stable economic expansion periods. Unlike KNN, the baseline probability rarely drops to zero. While the model does spike during actual recessions, these signals are obscured by the constant background noise. This pattern suggests overfitting—the model learned patterns in the training data that do not generalize to new observations.

5. Data Modeling

Three classification algorithms were implemented and compared: Logistic Regression, K-Nearest Neighbors (KNN), and Random Forest. Each brings different strengths and assumptions to the prediction task.

Model 1: Logistic Regression

Logistic Regression is a linear classifier that models the probability of recession as a function of the input features. It assumes a linear relationship between features and the log-odds of recession, making it interpretable but potentially limited in capturing complex patterns. For this problem, logistic regression serves as a baseline model against which more complex approaches can be compared.

Model 2: K-Nearest Neighbors (KNN)

KNN is a non-parametric algorithm that classifies new observations based on the majority vote of their k nearest neighbors in feature space. Unlike logistic regression, KNN makes no assumptions about the underlying data distribution. It works by asking: "When economic conditions looked similar to this in the past, were we in a

recession?" This instance-based approach is particularly intuitive for recession prediction, as it directly leverages historical analogies.

Model 3: Random Forest

Random Forest is an ensemble method that builds multiple decision trees and aggregates their predictions. Each tree is trained on a bootstrap sample of the data with a random subset of features, reducing overfitting and improving generalization. Random Forest also provides feature importance scores, offering insights into which economic indicators most influence predictions.

Validation Approach

Models were evaluated using a time-based train/test split rather than random cross-validation. This is critical for time series data—we cannot use future information to predict the past. The training set includes data from 1990 through approximately 2015, with the test set covering 2015-2025. This means the models are evaluated on their ability to predict the 2020 COVID recession and subsequent economic conditions.

Evaluation metrics include:

- **AUC-ROC:** Measures the model's ability to discriminate between recession and non-recession across all probability thresholds.
- **Precision:** Of all predicted recessions, what fraction were actually recessions? (Measures false alarm rate)
- **Recall:** Of all actual recessions, what fraction did we correctly predict? (Measures miss rate)
- **F1 Score:** Harmonic mean of precision and recall, balancing both concerns.

6. Results and Interpretation

Model Performance Comparison

Model	AUC-ROC	Precision	Recall	F1 Score
Logistic Regression	0.983	0.08	1.00	0.15
K-Nearest Neighbors	**0.987**	**0.29**	**1.00**	**0.44**
Random Forest	0.833	0.05	1.00	0.10

Table 2: Model Performance Metrics (Recession Class = 1)

Key Finding: All three models achieve perfect recall (1.00), meaning they successfully detected both recession months in the test set. However, they differ dramatically in precision. K-Nearest Neighbors emerges as the best performer with the highest AUC-ROC (0.987), best precision (0.29), and best F1 score (0.44).

Detailed Results by Model

Logistic Regression

Class	Precision	Recall	F1-Score	Support
0 (No Recession)	1.00	0.81	0.90	117
1 (Recession)	0.08	1.00	0.15	2

Table 3: Logistic Regression Classification Report

Logistic Regression produces a smooth, continuous probability output. The time-series visualization shows gradual probability increases leading into recessions. However, with a precision of only 0.08, roughly 92% of its recession predictions are false alarms. The model errs heavily on the side of caution—it would rather raise many false warnings than miss a single recession.

K-Nearest Neighbors

Class	Precision	Recall	F1-Score	Support
0 (No Recession)	1.00	0.96	0.98	117
1 (Recession)	0.29	1.00	0.44	2

Table 4: K-Nearest Neighbors Classification Report

KNN produces the cleanest signal of all three models. The probability stays near zero during stable economic periods and rises sharply only when conditions closely resemble historical recessions. This results in significantly fewer false alarms (precision of 0.29 vs 0.08 for logistic regression). The visualization shows nearly flat probability during non-recession periods with decisive spikes during actual recessions.

Random Forest

Class	Precision	Recall	F1-Score	Support
0 (No Recession)	1.00	0.70	0.82	117
1 (Recession)	0.05	1.00	0.10	2

Table 5: Random Forest Classification Report

Random Forest performed worst overall, with the lowest AUC-ROC (0.833) and precision (0.05). The visualization reveals why: the probability signal is noisy and erratic, fluctuating between 0.1-0.5 even during stable economic periods. This suggests the model is overfitting to training data patterns that do not generalize well.

Random Forest Feature Importance (Top 10)

Rank	Feature	Importance
1	us_gdp_growth_lag3	0.1785
2	us_gdp_growth_lag6	0.1349
3	us_yield_spread_lag12	0.0969
4	us_3m	0.0907
5	us_gdp_growth	0.0675
6	kr_gdp_growth	0.0641
7	uk_3m	0.0497
8	eu_3m	0.0465
9	jp_gdp_growth	0.0426
10	eu_yield_spread	0.0351

Table 6: Random Forest Feature Importance

Despite poor predictive performance, Random Forest's feature importance provides valuable insights. Lagged US GDP growth dominates (3-month and 6-month lags), followed by the 12-month lagged yield spread. International indicators also

appear—Korean GDP growth, UK and EU short-term rates, Japanese GDP growth—supporting the multi-country approach. However, a cautionary note: feature importance from a poorly performing model should be interpreted carefully.

The Precision-Recall Trade-off

A central finding is that all models achieve perfect recall but struggle with precision. This reflects the fundamental asymmetry of the problem. Given the severe consequences of missing a recession, models naturally lean toward sensitivity. However, a model that cries wolf too often loses credibility and practical utility.

In a real-world application, the acceptable false alarm rate would depend on context. Central bankers might tolerate more false positives to avoid ever missing a recession, while investors might prefer a more conservative model that signals only high-confidence warnings. KNN's balance—catching all recessions while reducing false alarms by roughly 70% compared to logistic regression—represents a reasonable middle ground.

Limitations

Several limitations affect interpretation of these results:

- **Small sample size:** With only three recessions in the dataset and two in the test set, statistical conclusions are inherently limited.
- **Unique recession characteristics:** Each recession had different causes (tech bubble, financial crisis, pandemic), making generalization difficult.
- **Recent yield curve anomaly:** The 2022-2023 yield curve inversion was the deepest on record but has not (yet) been followed by a recession, challenging traditional assumptions.
- **Data availability:** Some international series (e.g., China's 10-year bond yield) were unavailable through FRED, limiting the completeness of the multi-country analysis.

7. Code Structure

The project code is organized in a Jupyter notebook (main.ipynb) with modular sections and clear documentation. The structure follows best practices for reproducible data science.

Project Organization:

- **main.ipynb:** Primary analysis notebook containing all code, visualizations, and results
- **README.md:** Project documentation and setup instructions
- **requirements.txt:** Python dependencies for reproducibility
- **data/:** CSV and Excel files containing raw FRED data

Code Sections:

5. Data loading and database connection
6. Preprocessing and feature engineering
7. Exploratory data analysis and visualization
8. Model training (Logistic Regression, KNN, Random Forest)
9. Model evaluation and comparison

10. Results visualization and interpretation

Comments throughout the code explain the purpose, functionality, and reasoning behind each section. Variable names are descriptive, and magic numbers are avoided in favor of named constants where possible.

8. References

Data Sources

Federal Reserve Bank of St. Louis. (n.d.). *Federal Reserve Economic Data (FRED)*. https://fred.stlouisfed.org/

World Bank. (n.d.). *Global Economic Monitor* [Data set]. Data Catalog. https://datacatalog.worldbank.org/search/dataset/0037798/Global-Economic-Monitor

Background Reading

Estrella, A., & Mishkin, F. S. (1998). Predicting U.S. recessions: Financial variables as leading indicators. *Review of Economics and Statistics, 80*(1), 45-61.

National Bureau of Economic Research. (n.d.). *US business cycle expansions and contractions*. https://www.nber.org/research/data/us-business-cycle-expansions-and-contractions

Libraries and Tools

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

McKinney, W. (2010). Data structures for statistical computing in Python. *Proceedings of the 9th Python in Science Conference*, 51-56.

Nate's contributions and key lessons learned:

Contributions:

For this project, I helped with several different parts, mainly related to organizing the data, setting up the project, and creating slides.

- Collected and organized datasets from the Global Economic Monitor and FRED.

- I helped structure the folders in VS Code and made sure file names matched what our code expected.

- I worked on multiple presentation slides. I also helped review the graphs we generated and picked which ones to include.

Lessons I Learned:

- Economic data is messy and inconsistent.
    - Every country reports differently, and I learned how important resampling and filling techniques are.

    - Machine learning doesn't "predict the future," it estimates risk.
        - Our models could detect recession patterns but couldn't perfectly predict exactly when the next one will happen. This helped me understand why forecasting real-world events is challenging.

- Working in a group with GitHub is valuable.
    - I learned how important it is to keep the project organized and communicate clearly when multiple people are working in the same notebook and dataset.

Julian's contribution and key lessons learned:

For this project, I implemented three different classification models to predict economic recessions: Logistic Regression, Random Forest, and K-Nearest Neighbors (KNN). The goal was to see which model could best identify recession periods using economic indicators from six major economies.

For the Logistic Regression model, I used scikit-learn's LogisticRegression class with balanced class weights since recessions are rare events. The model was trained on

scaled features using StandardScaler to normalize the data. I set the threshold at 0.5 for predictions and evaluated it using AUC-ROC, precision, recall, and F1 scores.

The Random Forest model was a bit trickier. I started with 200 trees but the model kept overfitting on the training data. I tried reducing the number of trees to 50 and limiting the depth to 5, as well as increasing the class weight for recessions to 50. Even with all these adjustments, the best precision I could get was 0.05 which honestly wasn't great.

KNN ended up being the best performer. I used KNeighborsClassifier with 5 neighbors and it worked really well for this problem. My thinking is that recession prediction benefits from pattern-matching to historical conditions rather than learning complex rules from limited examples. KNN basically asks "have we seen this before?" which makes sense for economic forecasting.

For the visualizations, I created three-panel plots for each model using matplotlib. The top panel shows the model's recession probability over time with actual recessions shaded in red. The middle panel displays the US yield curve spread with inversions highlighted since that's a known recession indicator. The bottom panel shows GDP growth as a bar chart with green for positive and red for negative growth. I used mdates for formatting the x-axis by year and saved each plot at 150 dpi.

Key Takeaways:

The biggest thing I learned is that simpler models can outperform complex ones when data is limited. We only had about 26 recession months in the training set and 2 in the test set, so Random Forest didn't have enough examples to learn meaningful patterns. KNN achieved the best balance with 0.987 AUC-ROC, perfect recall, and 0.29 precision.

I also learned that low precision scores are realistic for recession prediction - it's not a model failure. Since recessions are rare, even a good model will have some false alarms. The important thing is that all three models caught every recession (recall of 1.00), they just differed in how many false positives they generated.

Finally, I realized that feature importance scores from Random Forest were essentially meaningless in this case since the model wasn't actually predicting recessions correctly despite good training metrics. This taught me to always validate what the model is actually doing rather than just trusting the numbers.

Ben’s contribution and key lessons learned:

Contributions:

For this project, I helped with several different parts, deciding which models, project structure, project approach, preprocess and organize data

- I collected, organized, and chose datasets from the Global Economic Monitor and FRED.

- Preprocess all data from csv/xlsx to a dataframe.
- Set up the SQL database and environment.
- I chose which features to include in the model.
- I decided our approach, including which indicators to use, what years to include, and which models (Logistic Regression, Random Forest, KNN).
- I worked on presentation slides.
- I tested and debugged throughout the project.

Lessons I Learned:

- Some countries' data was not available, or some did not reach as far as U.S. economic data because some countries were still developing in the 1900s.
- Some machine learning models are better than others at predicting recessions.
- Recessions are onerous to predict accurately because of many more constantly changing variables.
- Preprocessing is very labor intensive.
- Using different models provides a bigger picture because different models excel at different things. Such as, KNN can find similarities among recessions, but can’t actually learn recessions.