Societal Unrest Data Analysis Report

Date: June 19, 2025
Author: Vishnu Panganamamula

Summary

The objective of this challenge was to develop a machine learning model that successfully can predict possible unrest events monthly economic, political, environmental and other indicators. My method was to test two machine learning models to cross-check and see how both perform to determine which model performed better for this dataset. The models that I used were a logistic regression model and a random forest classifier decision tree to train the data on. What was most important in the data analysis was the recall of the models. The recall initially was not up to standards, and after calibration, weight balancing, and balancing the dataset, the recall and other metrics for both models improved significantly.

Methodology

I wanted to train two models to understand how each model will interpret the data and to see if one model outperforms the other after training. The two models that I wanted to use were a logistic regression model and a random forest classifier decision tree. The reason that I picked these two models was that both models can determine binary probabilities, and the unrest.csv file has the appropriate data and target variable for these models in particular.

Section 1: Initial Model Training

I wanted to first upload the dataset into the Google Colab environment and see what the data is like and what features are in the dataset. I uploaded the dataset as a pandas dataframe through a raw GitHub link so that the data is automatically loaded into the testing environment instead of having to download the dataset each time or mounting my Google Drive into the Colab environment.

The next step was to standardize the dataset. Standardization is a key step in preprocessing and will allow for the models to understand the data better due to features being more uniform and prevents one feature being more influential over another when it comes to training. Once the models are initialized, the next step is to train and evaluate their results.

Section 2: Improvements to Models and Performance

Once I got the results for the initial models, I noticed that there needed to be more improvements to be made in order to improve the model performance. Before training the new models I first wanted to take a look at the dataset again. I saw that the dataset was unbalanced, especially for the target variable, the unrest_event. More cases were “No unrest” (0) rather than “unrest” (1). This made the results of the initial models underwhelming.

I first set out to balance the dataset using SMOTE (Synthetic Minority Oversampling Technique), to create more synthetic data of unrest_events to balance the target variable. This would be useful for the random forest model.

Next, I used some feature engineering techniques to create different features that can better predict unrest events in the future. For example, I created two features, economic_stress (unemployment_rate multiplied with the inflation_rate columns) and social_strain, which is a multiplication of gini_index and food_price_index. Since the dataset consists of the date when the unrest event has happened, I introduced a temporal feature which is the average of unrest events every 3 months, called unrest_3month_avg. The idea here was that these new features alongside the given features in the dataset will improve model metrics and performance.

As with the train-test split in Section 1, there is also a train-test split in Section 2, but this was with temporal data. I did an 80/20 split of the last 20% of time periods in the dataset so that the models can predict future events more accurately.

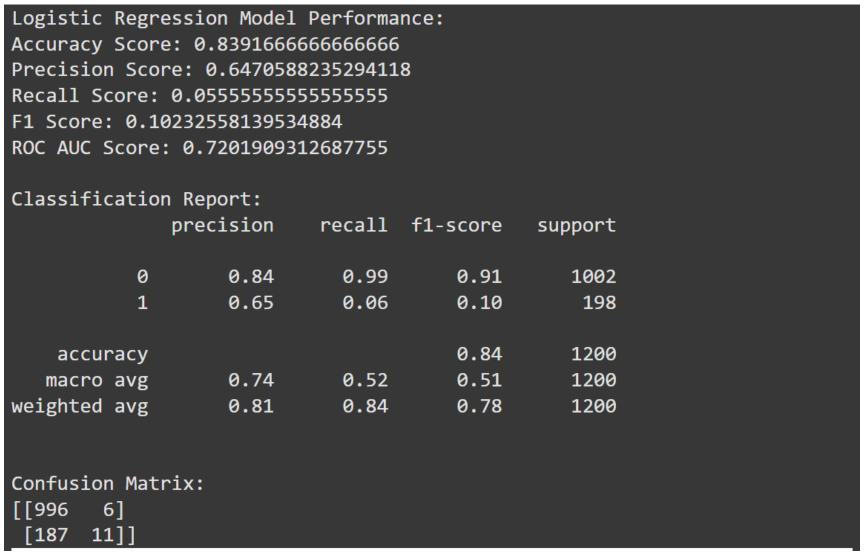
Once the preprocessing and feature engineering section of Section 2 was done, the next step was to initialize the models. Both models were created with balanced weights, so that one feature does not dominate over the other. The next step was to evaluate the models' performance and prediction capabilities. This time, a default threshold and optimal threshold was added to the evaluation metrics to see how well each model performs in each case.

I also added a feature importance graph for each model to see which features in the dataset (after feature engineering) were more important for predictions.

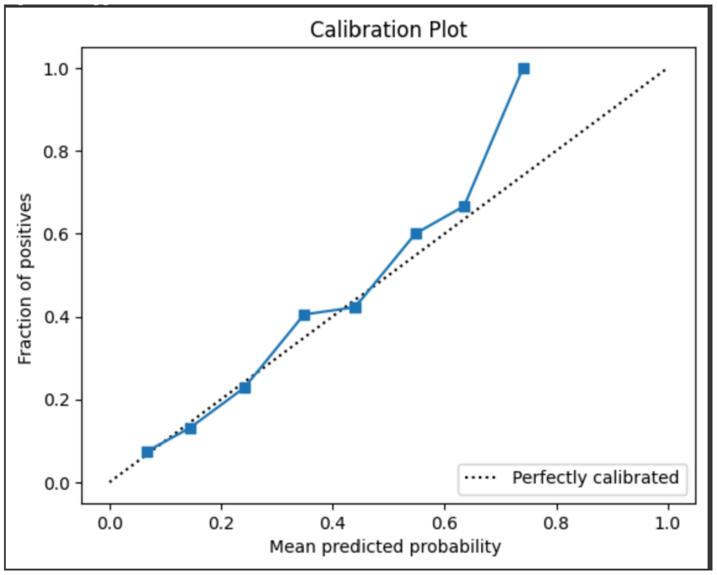
Results and Discussion

Section 1: Results

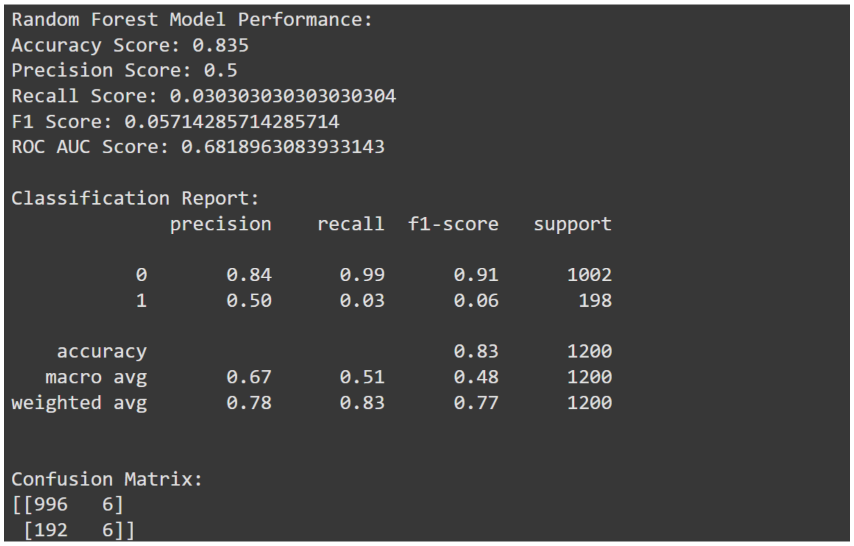
Logistic Regression



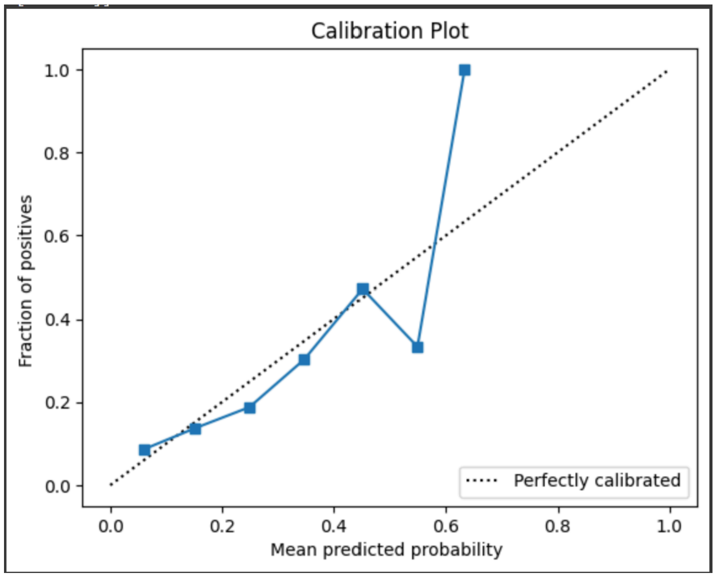
```
Logistic Regression Model Performance:
Accuracy Score: 0.8391666666666666
Precision Score: 0.6470588235294118
Recall Score: 0.05555555555555555
F1 Score: 0.10232558139534884
ROC AUC Score: 0.7201909312687755

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.99      0.91      1002
           1       0.65      0.06      0.10       198

    accuracy                           0.84      1200
   macro avg       0.74      0.52      0.51      1200
weighted avg       0.81      0.84      0.78      1200


Confusion Matrix:
[[996   6]
 [187  11]]
```



Random Forest

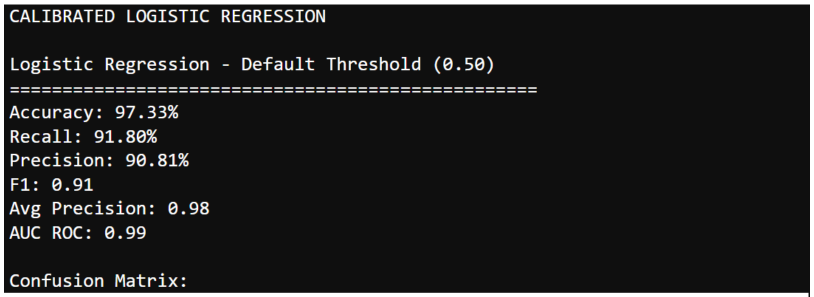


```
Random Forest Model Performance:
Accuracy Score: 0.835
Precision Score: 0.5
Recall Score: 0.030303030303030304
F1 Score: 0.05714285714285714
ROC AUC Score: 0.6818963083933143

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.99      0.91      1002
           1       0.50      0.03      0.06       198

    accuracy                           0.83      1200
   macro avg       0.67      0.51      0.48      1200
weighted avg       0.78      0.83      0.77      1200


Confusion Matrix:
[[996   6]
 [192   6]]
```

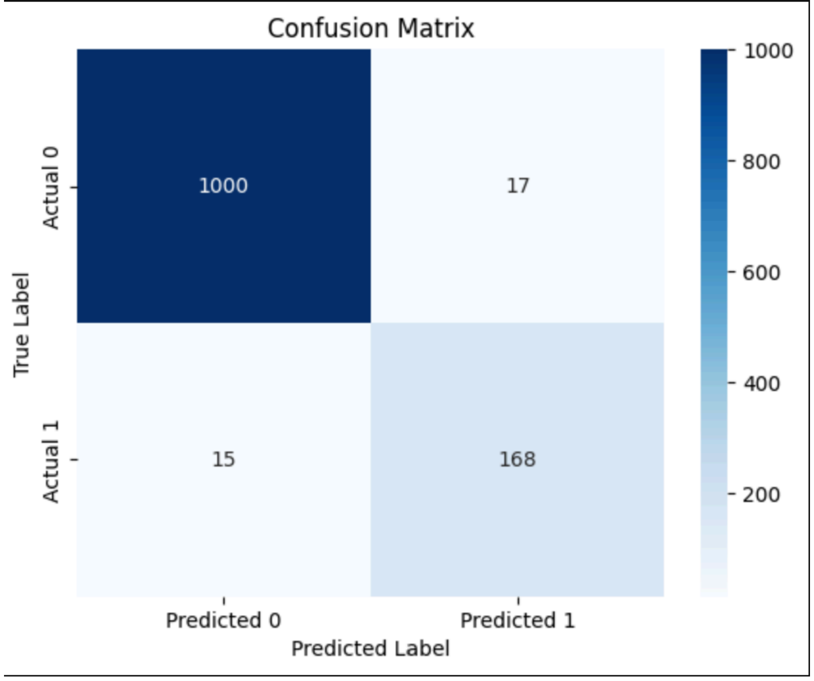


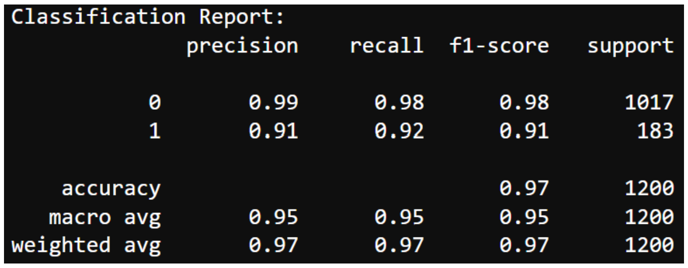
Section 2: Improvements to Models and Performance

Logistic Regression: Default Threshold and Optimal Threshold

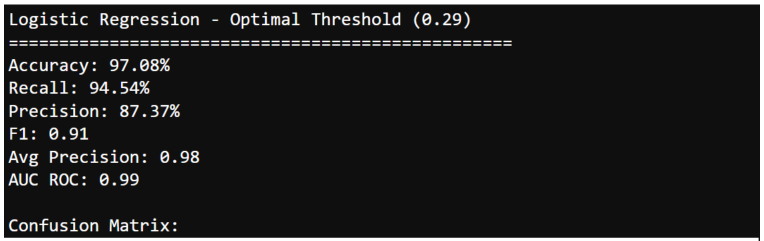


```
CALIBRATED LOGISTIC REGRESSION

Logistic Regression - Default Threshold (0.50)
==================================================
Accuracy: 97.33%
Recall: 91.80%
Precision: 90.81%
F1: 0.91
Avg Precision: 0.98
AUC ROC: 0.99

Confusion Matrix:
```

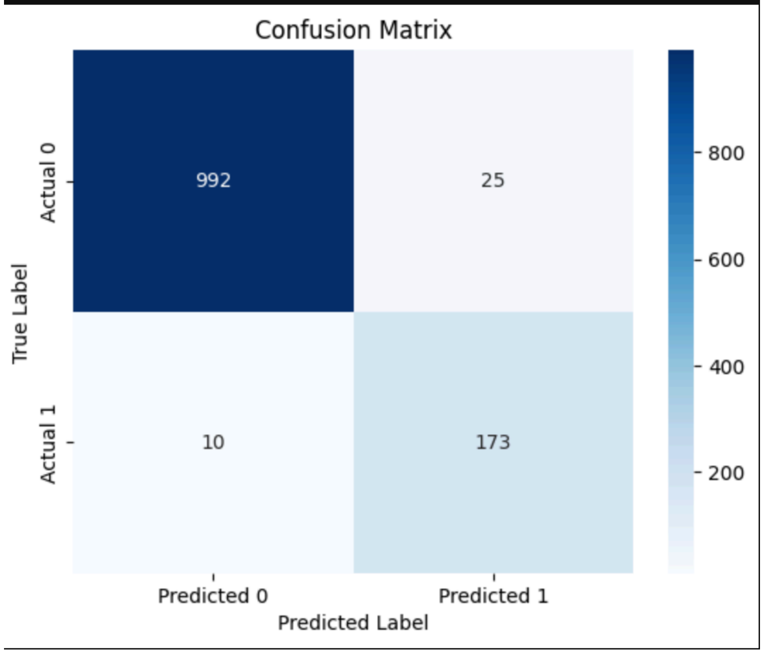


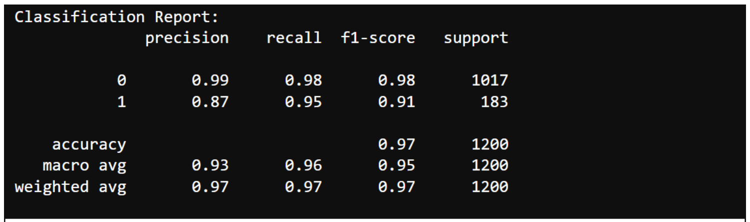


```
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      1017
           1       0.91      0.92      0.91       183

    accuracy                           0.97      1200
   macro avg       0.95      0.95      0.95      1200
weighted avg       0.97      0.97      0.97      1200
```

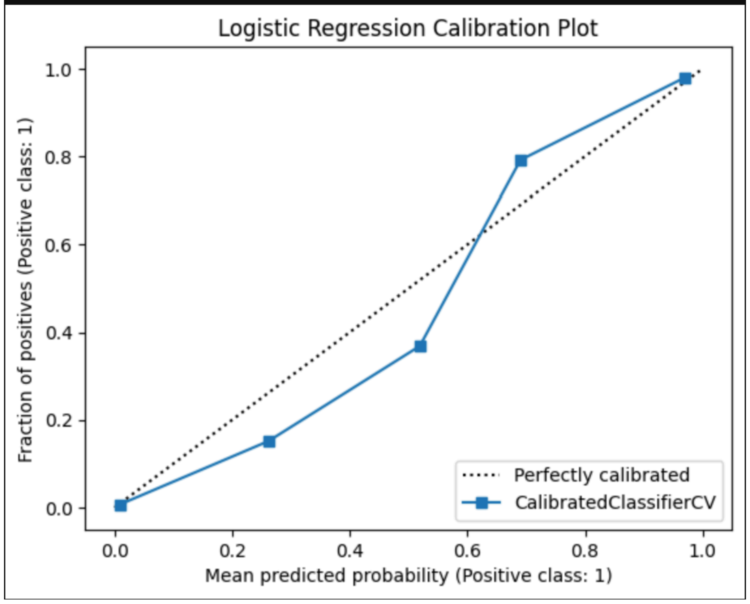


```
Logistic Regression - Optimal Threshold (0.29)
==================================================
Accuracy: 97.08%
Recall: 94.54%
Precision: 87.37%
F1: 0.91
Avg Precision: 0.98
AUC ROC: 0.99

Confusion Matrix:
```

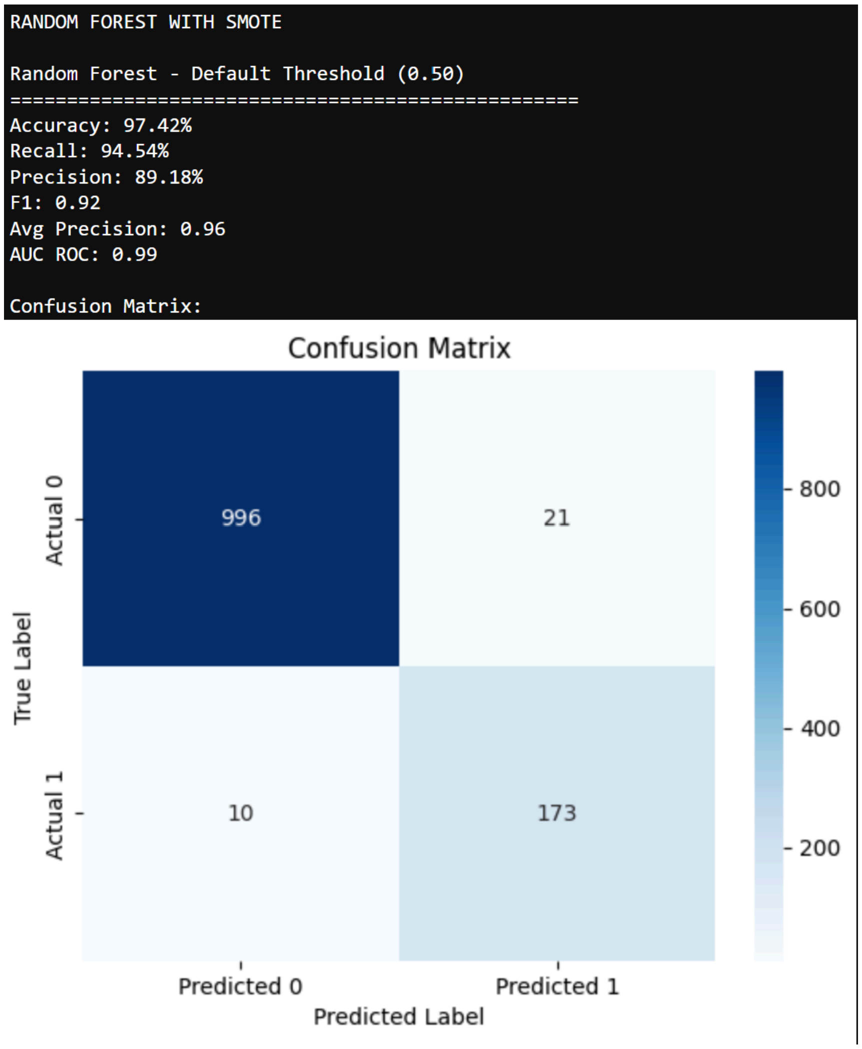




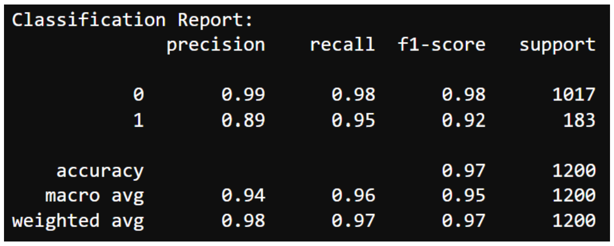
```
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      1017
           1       0.87      0.95      0.91       183

    accuracy                           0.97      1200
   macro avg       0.93      0.96      0.95      1200
weighted avg       0.97      0.97      0.97      1200
```



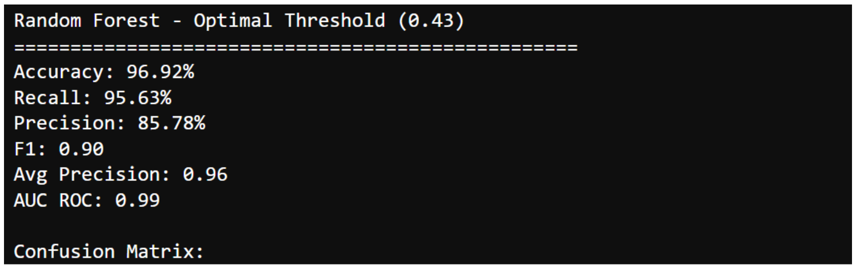
Random Forest: Default Threshold and Optimal Threshold



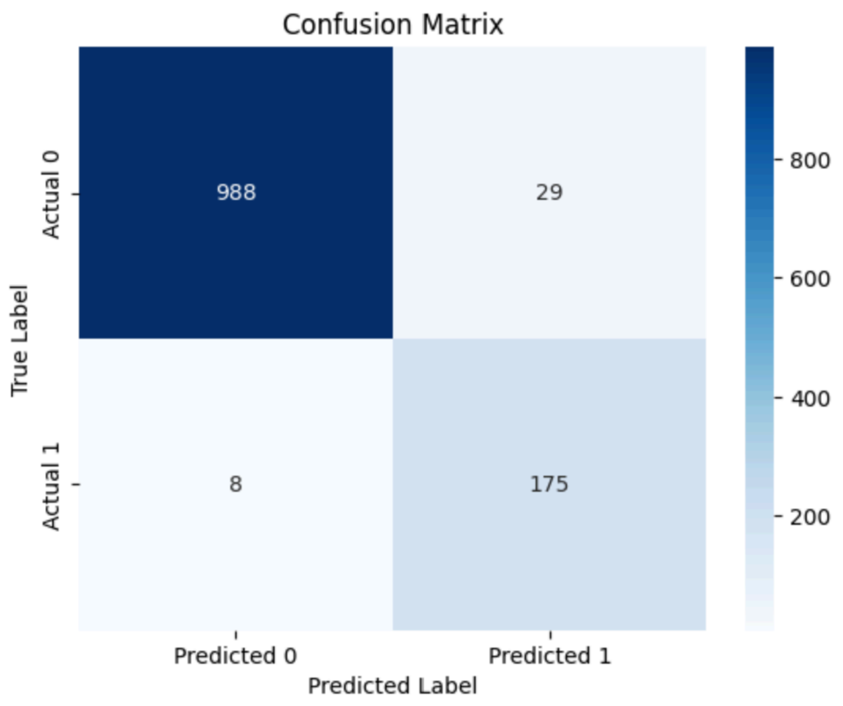
```
RANDOM FOREST WITH SMOTE

Random Forest - Default Threshold (0.50)
==================================================
Accuracy: 97.42%
Recall: 94.54%
Precision: 89.18%
F1: 0.92
Avg Precision: 0.96
AUC ROC: 0.99

Confusion Matrix:
```

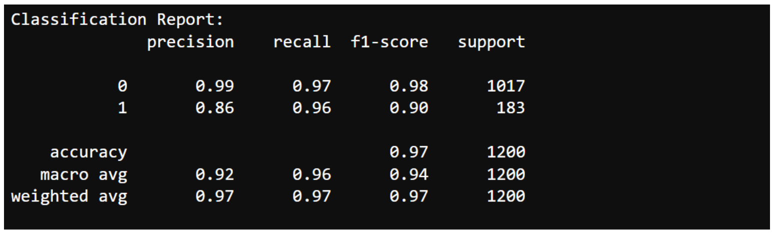


```
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      1017
           1       0.89      0.95      0.92       183

    accuracy                           0.97      1200
   macro avg       0.94      0.96      0.95      1200
weighted avg       0.98      0.97      0.97      1200
```

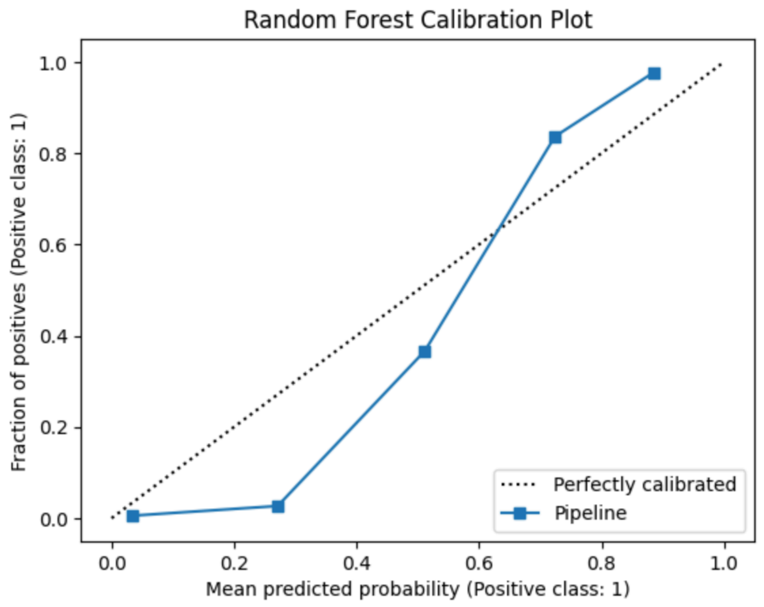


```
Random Forest - Optimal Threshold (0.43)
==================================================
Accuracy: 96.92%
Recall: 95.63%
Precision: 85.78%
F1: 0.90
Avg Precision: 0.96
AUC ROC: 0.99

Confusion Matrix:
```

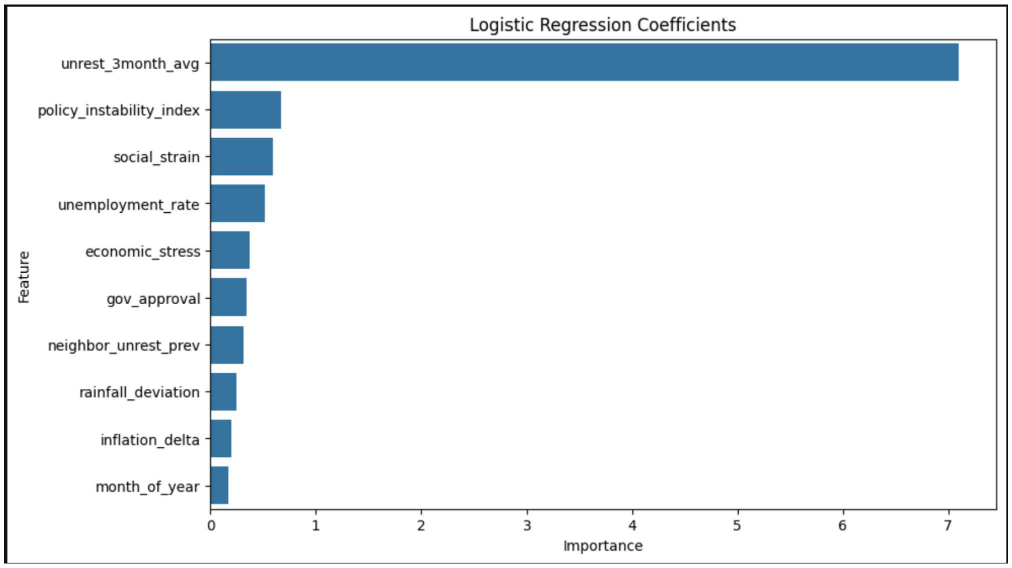


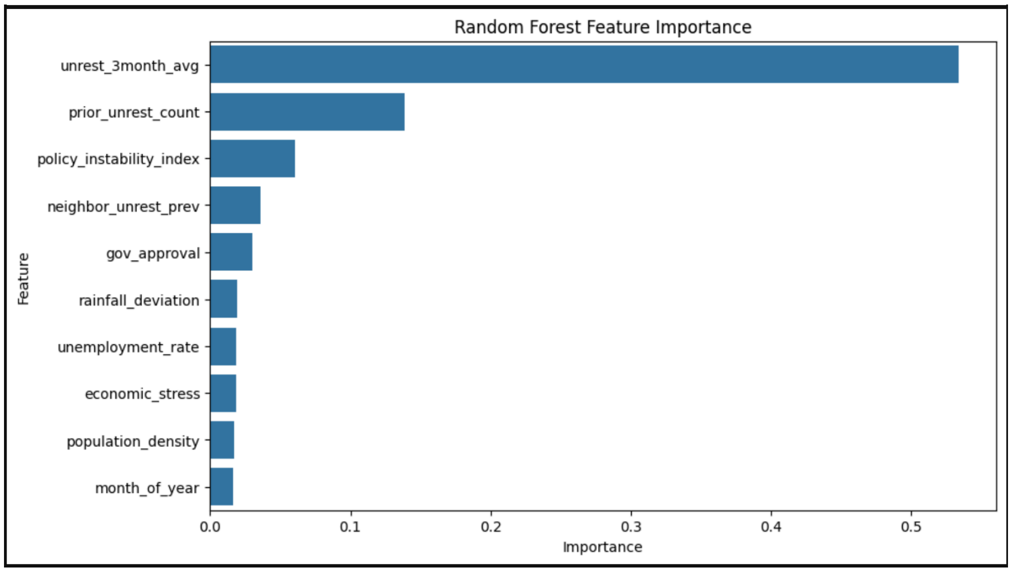


```
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      1017
           1       0.86      0.96      0.90       183

    accuracy                           0.97      1200
   macro avg       0.92      0.96      0.94      1200
weighted avg       0.97      0.97      0.97      1200
```



Feature Importance Graphs of Logistic Regression and Random Forest





Based on the results, the logistic regression and random forest both improved in Section 2 compared to Section 1. In Section 1, while the overall accuracy for both models was somewhat high, the other statistics were extremely low, particularly the recall rate. It was 5.6% and 3.0% for the logistic regression and random forest models respectively. This means that the models were only able to identify aforementioned percentages of unrest events (1), but were not able to identify more than that. Moreover, in the confusion matrix, only 11 true positive cases and 6 true positive unrest events were detected by the logistic regression and random forest respectively. There were more false negative and false positive cases recorded by both models than true positive cases. This can be directly observed because of the low recall rate for both models.

In Section 2, the improvements were noticeable in both models. To start, all metrics improved greatly for each model. To start, the logistic regression model was able to record a 97% general accuracy rate with the recall rate being 91.8% for the default threshold (0.50). As for the optimal threshold (0.29), the recall rate improved to 95%. This means that the model now predicts up to 95% of unrest cases accurately from 5.6%. The confusion matrices for both thresholds also now show more true positive and true negative cases and less false positive and false negative cases, meaning the model is now more reliable.

As for the random forest, in the default threshold (0.50) and the optimal threshold (0.43), the recall rate was around 95% in both cases. This is another massive improvement for the model prediction capability from the 3% recall in the initial cases. As for the calibration plots, both models in Section 2 are closer to the ideal calibration line of best fit. The calibration, feature engineering and time series splits which augmented the dataset were the reasons for the predictions to improve, as well as weight balancing for the models.

Moreover, in the feature importance graphs, it can be observed that the unrest_3month_avg and policy_instability_index features are the most important features for both models to come up with predictions on future unrest events. However, logistic regression places more importance with social strain, unemployment and economic stress, while the random forest places more importance on neighbor unrest, government approval, prior unrest situations for some regions.

In conclusion, any policy analysts should use the optimal threshold models since they detect more true positive cases and actually predict more true risks in unrest events happening, and have more manageable false alerts (low percentage of false alert cases for both models).

Limitations and Future Work

I feel like the only limitation with this project was with the dataset itself. While it captures unrest events over a 10 year period and for every month, I think the dataset would be more comprehensive if it also included particular days when an unrest event happened instead of just the first day of each month for each year. I think that is the only limitation to the dataset. However, even with that extra data, the next challenge would be to properly preprocess and analyze that data for adequate predictions.

Appendix

- GitHub Repository Link: https://github.com/vishnupan819/unrest_data