Political Alignment Quiz and Machine Learning Model

This report outlines the design and implementation of a political alignment quiz application developed in C++, alongside a machine learning model in Python that analyzes collected user responses to improve prediction accuracy over time. The system combines interactive survey logic with data-driven insights to identify a user's political leaning based on their answers to a series of questions.

# Rationale for the Chosen ML Model

The machine learning model selected for this project is a Random Forest Classifier. This choice was motivated by several factors: Random Forests are robust to overfitting, handle categorical and numerical data well, and provide high accuracy even with relatively small datasets. They also offer feature importance metrics, which can help identify which survey questions are most influential in predicting political alignment. Given the nature of the input data—numerical scores assigned to political parties based on user responses—Random Forests are well-suited to capture complex patterns and interactions. (McGuire & Delahunt, 2020)

# Data Storage and Processing

User responses are stored in a CSV file named 'responses.csv'. Each row in the file represents a single user's interaction with the quiz, including their cumulative scores for each political party, the predicted party based on those scores, and the user's self-identified political label. This structured format allows for easy ingestion into the Python-based machine learning pipeline. The Python script reads the CSV using pandas, maps party labels to numeric values, and splits the data into training and testing sets. The features used for training are the party scores, and the target variable is the user's actual political label.

# Improving Predictions Over Time

As more users complete the quiz and their responses are appended to the CSV file, the dataset grows, enabling the model to learn from a broader range of inputs. The training script can be rerun periodically to retrain the model on the updated dataset, thereby improving its predictive accuracy. This iterative learning process ensures that the model adapts to changing patterns in user responses and becomes more reliable over time. Additionally, the use of a Random Forest allows the model to generalize well from limited data, making it effective even in early stages of deployment.

# Evaluation of Model Accuracy and Limitations

The model's performance is evaluated using standard classification metrics: accuracy, precision, recall, and F1-score. These metrics provide a comprehensive view of how well the model predicts political alignment. Initial tests show promising results, with accuracy and F1-scores typically above 80%, depending on the size and diversity of the dataset. However, the model has limitations. It may struggle with ambiguous or neutral responses that do not strongly favor any party. Additionally, if the dataset is imbalanced—i.e., more users identify with one party than others—the model may become biased. To mitigate this, techniques such as stratified sampling or class weighting can be employed during training. (Siddique et al., 2023)

# Conclusion

This project demonstrates the effective integration of a C++ survey application with a Python-based machine learning model. By collecting user responses and continuously training the model, the system evolves to provide increasingly accurate predictions of political alignment. The use of Random Forests ensures robustness and interpretability, while the structured data storage facilitates seamless analysis. Future enhancements could include real-time prediction using a saved model, dynamic question loading, and deployment as a web application.

# References

McGuire, S. K., & Delahunt, C. B. (2020). Predicting United States Policy Outcomes with Random Forests. Institute for New Economic Thinking. https://www.ineteconomics.org/uploads/papers/McGuire-and-Delahunt-predictingPolicy\_INET\_25oct2020.pdf

Siddique, S., et al. (2023). Survey on Machine Learning Biases and Mitigation Techniques. Machine Learning and Knowledge Extraction, 4(1), 1. https://www.mdpi.com/2673-6470/4/1/1