CS 109a: Coup d’Etat Project

By: Andrew Nickerson, Caleb Saul, Ayana Yaegashi, Drew Webster

Motivation, Context & Framing of the Problem

Motivation & Context:

Coups are important and impactful events which still occur frequently in today’s day and age. Just recently, there was an attempted coup in Peru in which the president attempted to dissolve congress and re-write the country’s constitution. Over the summer, the Japanese former prime minister was assassinated. These political events have the capacity to change the course of a country’s history as well as the lives of the people who live in that country. As a result, we feel that it is important to understand coups better from a historical perspective and try to better understand what causes coups and coup-like events to occur.

Problem:

Through our analysis of the data we hope to answer the following two questions:

- *Can we predict whether a coup event will be successful (realized) based on some set of predictors? If so, which predictors are most useful for this task?*
- *Can we predict if an event was a conspiracy, an attempted coup, or an actual coup? If so, which predictors are most useful in this scenario?*

Data Description and How it was Handled

The Cline Center Coup D’état Project (CPD) Dataset "identifies coups, attempted coups, and coup plots/conspiracies in 136 countries (1945-2019). The data identifies the type of actor who initiated the coup (i.e.military, palace, rebel, etc.) as well as the fate of the deposed executive (killed, injured, exiled, etc.)."[1] The data is tabular with the following variables present:

Variable	Description
coup_id	a unique number assigned to each event. It consists of the country’s cowcode and the eight digit date of the event in MMDDYYYY
cowcode	a unique country code number used to identify the country where a coup event occurred
country	name of the country where the coup event occurred
year	year of the coup event
month	month of the coup event
day	day of the coup event
event_type	indicates whether the event is a coup, attempted coup, or conspiracy

[1] Peyton, Buddy; Bajjalieh, Joseph; Shalmon, Dan; Martin, Michael; Bonaguro, Jonathan (2021): Cline Center Coup D’état Project Dataset. University of Illinois at Urbana-Champaign. https://doi.org/10.13012/B2IDB-9651987_V3

Variable	Description
realized	a dummy variable where one indicates a successful coup and zero otherwise.
unrealized	a dummy variable where one indicates an unsuccessful coup or plot and zero otherwise
conspiracy	a dummy variable where one indicates a coup conspiracy thwarted prior to execution and zero otherwise
attempt	a dummy variable where one indicates a coup was attempted but failed and zero otherwise
military	a dummy variable where one indicates a military coup/attempt/conspiracy and zero otherwise
dissident	a dummy variable where one indicates a dissident coup/attempt/conspiracy and zero otherwise
rebel	a dummy variable where one indicates a rebel coup/attempt/conspiracy and zero otherwise
palace	a dummy variable where one indicates a palace coup/attempt/conspiracy and zero otherwise
foreign	a dummy variable where one indicates a foreign-backed coup/attempt/conspiracy and zero otherwise
auto	a dummy variable where one indicates an auto coup and zero otherwise
resign	a dummy variable where one indicates a forced resignation and zero otherwise
popular	a dummy variable where one indicates a popular revolt and zero otherwise
counter	a dummy variable where one indicates a counter-coup and zero otherwise
other	a dummy variable where one indicates the coup event does not fit into any of the above categories or the actors were not identified and zero otherwise
noharm	a dummy variable where one indicates the deposed executive was not harmed during the coup event and zero otherwise
injured	a dummy variable where one indicates the deposed executive was injured during the coup event and zero otherwise
killed	a dummy variable where one indicates the deposed executive was killed during the coup event and zero otherwise
harrest	a dummy variable where one indicates the deposed executive was placed under house arrest and zero otherwise
jailed	a dummy variable where one indicates the deposed executive was jailed and zero otherwise
tried	a dummy variable where one indicates the deposed executive was tried and zero otherwise
fled	a dummy variable where one indicates the deposed executive fled the country and zero otherwise
exile	a dummy variable where one indicate the deposed executive was banished from the country and zero otherwise

This data was gathered by trained and vetted student analysts who synthesized data from the following sources into this single dataset:

- The Center for Systemic Peace (Marshall and Marshall, 2007)
- The World Handbook of Political and Social Indicators (Taylor and Jodice, 1983)
- Coup d’état: a Practical Handbook (Luttwak, 1979)
- The Cline Center’s Social, Political, and Economic Event Database (SPEED) Project

- Government Change in Authoritarian Regimes project - 2010 Update (Svolik and Akcinaroglu, 2006)
- Powell and Thyne's Coup Data (2011)
- News articles from ProQuest's Historical Newspapers collection
- Wikipedia

Events were included in the dataset under the condition that they were sufficiently sourced (at least two sources) and lacked inconsistencies (no irreconcilable event dates).

See CDP Dataset Codebook for further details on student analyst training and data quality checks.

Exploratory Data Analysis

Upon inspection of the data set, we saw that we had 943 observations with 29 columns. The columns **cowcode, coup_id, event_type,** and **unrealized** were dropped due to the fact that their data was already encoded in other columns and were therefore unnecessary. We are also dropping the columns corresponding to executive fate – 'noharm','injured' , and 'killed' – and executive consequence – 'harrest','jailed' ,'tried','fled', and 'exile'. Because we are building a model to classify coup events, using these variables as predictors, when they can be seen as proxies for the event itself, would give our model an unfair and unrealistic advantage.

We then found an error in the data, where an observation in Nigeria had a day of 0, which is impossible. This observation was dropped from the dataframe to prevent the use of incorrect/corrupt data.

Since we have far too many unique countries, years, months, and days for us to be able to use one-hot-encoding to represent the values, we transform **countries to continents, years to decades, months to seasons**, and **days to timeframe within the month** (early/middle/late).

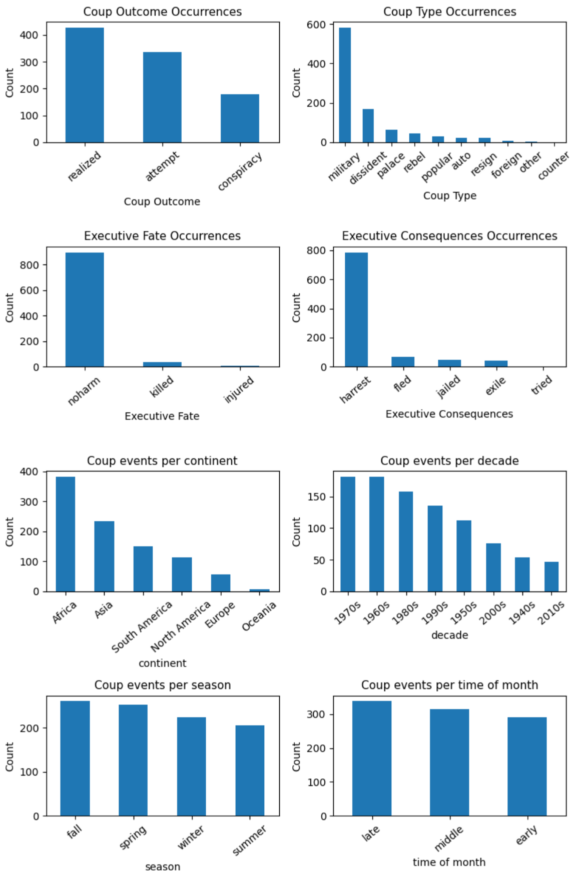


Figure 1 *(right) shows a plot giving the distribution of values for several variables. Note that coup types are not mutually exclusive, so a coup could be counted as more than one type. We observe that the majority of executive fates are* ***no harm****, with the majority of executive consequences being* ***house arrest****. In addition, most coups occurred in* ***Africa*** *in our dataset. The decades of the coups are skewed left, with most coups happening in the* ***1960s****.*

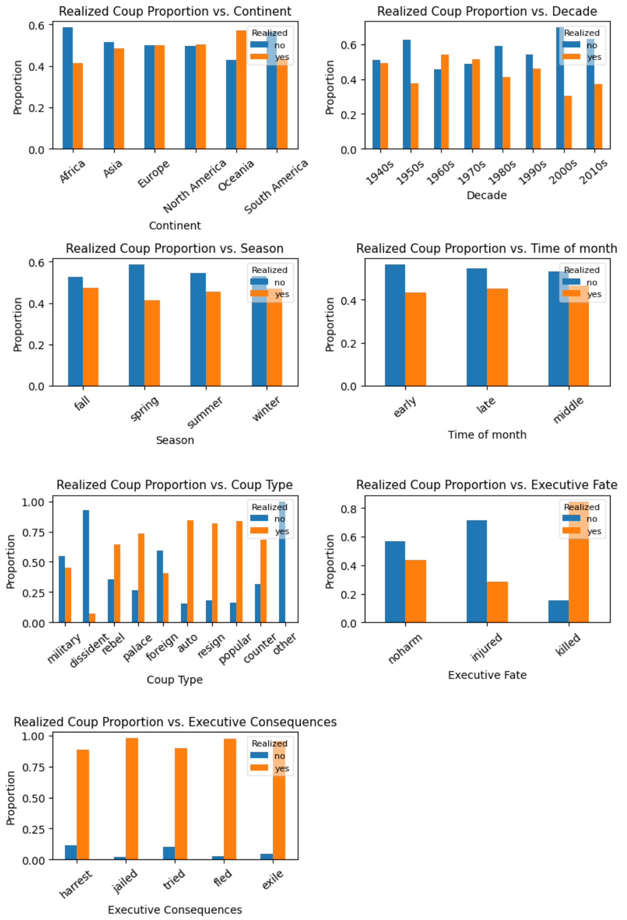
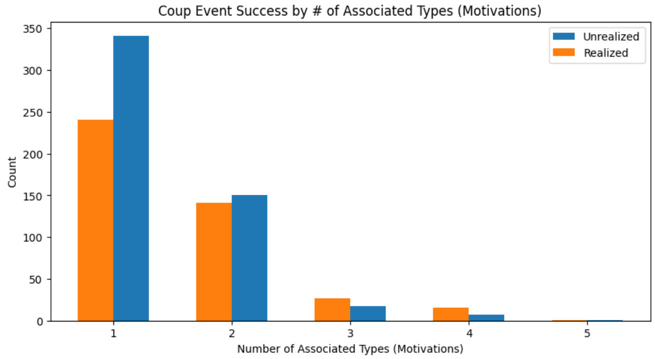


Figure 2 (left) shows relations between each variable and whether or not each coup was realized. We do not notice any particularly strong indicators; however we observe that the ***majority of 1960s and 1970s coups in our dataset were realized****, while the majority of coups from other decades were not. Additionally, we observe that the* ***majority of military, dissident, and foreign coups were not realized****, while the majority of other coup types were realized. We also see that* ***a significantly greater proportion of coups <u>were not</u> realized in Africa,*** *while a* ***significantly greater proportion of coups <u>were</u> realized in Oceania.*** *Finally, we observe that* ***virtually all coups that ended in killing the executive were realized****.*

We calculated that the proportion of realized vs. unrealized coup events in our data is 45.17% vs. 54.83% respectively.

Figure 4 *(right) plots whether a coup is realized or not grouped by the number of types associated with the event (many data points have more than one coup "type" listed). We see that there seems to be an association between the number of associated types/motivations and whether the coup is successful;* ***the more motivations, the higher the proportion of realized coup events****.*



Modeling Approach

We decided to create two classes of models: those that predict the binary of whether or not a coup was realized (predicting "realized"), and models that predict between the three classes of coup type, or "event type," which includes the classes "realized", "attempt", and "conspiracy". For each prediction goal, we created a **baseline logistic regression model with no regularization penalty**, a **logistic regression model with Lasso penalty**, a **single decision tree model**, a **bagging tree model**, a **random forest tree,** an **AdaBoost model** and a **Gradient descent boosting model**.

In order to create the logistic regression and tree models, we turned to one-hot-encoding most predictors of our data. For example, instead of using the "continent" column, we encoded this into one column for Africa, one for South America, etc., with 0/1 indicating whether that observation is from that continent.

Also, in order to address the balance problem (rather than dropping data), we stratified our train-test split so that equal proportions of each predicting class (realized vs. unrealized or coup vs. attempt v.s conspiracy) are represented in both the train and test set.

Predicting Realized and Unrealized Coups

Logistic Regression Model

Our initial baseline model classified between the two classes: realized and unrealized. We decided to use a logistic regression model for our baseline because we wanted a simple model that performs well when predicting between classes.

We have many predictors (23) in our data set after one-hot encoding, so we decided to use a logistic regression **with Lasso penalty** model next to eliminate coefficients for less important predictors. We used cross validation to identify the best regularization strength (1.0) and then used bootstrapping to further evaluate the importance of each predictor. Any predictor whose 95% confidence interval after bootstrapped included 0 was deemed unimportant.

Tree Models

Based on our research, we found that tree models were likely the best kind of model to use given the categorical nature of our data. We began using a **single decision tree** as a baseline tree model. Then, we wanted to try out different tree model optimization techniques, so we created a tree model using **bagging**, **random forest**, **AdaBoost**, and **gradient descent** and evaluated the results. We consistently used a random state of 109 for these models. In addition, for the bagging and random forest models we used cross validation to find the optimal number of estimators as well as the optimal maximum depth of the base estimator model.

We trained and evaluated each of these models, and compared accuracy scores between models. See **Results** section for details on these evaluations.

Predicting between Multiple Classes: Realized, Attempted, or Conspiracy

Logistic Regression Model

Building off of the original question, we wanted to expand the scope of our model to predict what type of coup even occurred rather than if one was successful or not (ie Realized, Attempted, or Conspiracy).

Similarly to the binary class case, we started off with a Logistic Regression model, this time multinomial, and then used a **Lasso penalty** with the **saga solver** to zero out coefficients of the least important predictors using cross validation to get the best regularization strength (1.0). Bootstrapping was used to evaluate the importance of predictors.

Tree Models

We turned our tree models to the multiple class case to see how they would compare to the logistic regression. Again we started with a **single decision tree** as a baseline and then moved on to various ensemble methods to compare (the same methods as above for binary-predicting models).

We trained and evaluated each of these models, and compared accuracy scores between models. For these models, since we were predicting between multiple classes, we also found the accuracy scores ***by class*** and compared these between models. See **Results** section for details on these evaluations.

Results, Conclusions, Strengths & Limitations, and Future Work

Results:

Predicting Realized and Unrealized Coups

A summary of our models' train and test accuracy is below:

	train	**test**
Baseline Logistic Regression	0.7384	0.7619
Lasso Logistic Regression	0.7517	0.7460
Single Decision Tree Depth = 5	0.7623	0.6931
Bagging	0.7530	0.7302
Random Forest	0.7968	0.7460
Adaboost	0.7676	0.7355
Gradient Boosting	0.7543	0.7249

Logistic Regression Model

For our logistic regression model with Lasso penalty, we found the following predictors were least important (zeroed out by Lasso): **`Africa, Asia, military, foreign,`** and **`other.`**

After bootstrapping, we had a better idea as to which predictors were most important. We note that the following coefficients did not include 0 in a 95% confidence interval after 100 bootstraps, sampling with replacement from our training data (753 data points): **`North America, Oceania, 1960s, 2000s, dissident, palace, auto, resign, popular, counter,`** and **`other.`** There is little overlap between these "non important" features from bootstrapping and the important features from Lasso regression, which confirms our list of important features.

Tree Models

For our tree models, according to permutation importance the following predictors were the top 3 best predictors: **`dissident, military, popular,`** for every model with dissident being the best by far.

Meanwhile, **`auto`** was the worst for the bagging model; **`early, North America, South America,`** and **`spring`** were the worst for the random forest. **`Europe, auto`** and **`fall`** for Adaboost and **`auto`** and **`fall`** for the Gradient Boost

Predicting between Multiple Classes: Realized, Attempted, or Conspiracy

A summary of our models' train and test accuracy is below:

Model	**train**	**test**
Baseline Logistic Regression	0.6401	0.6032
Lasso Logistic Regression	0.6401	0.6032
Single Decision Tree Depth = 5	0.6428	0.5714
Bagging	0.7184	0.5661
Random Forest	0.6865	0.5820
Adaboost	0.6215	0.5873
Gradient Boosting	0.6401	0.6031

We also broke down our accuracy results by class, as follows:

Model	**coup:** train	**coup:** test	**attempted:** train	**attempted:** test	**conspiracy:** train	**conspiracy:** test
Baseline Logistic Regression	0.7765	0.8023	0.5130	0.4627	0.5556	0.3889
Lasso Logistic Regression	0.7765	0.8023	0.5130	0.4627	0.5556	0.3889
Single Decision Tree Depth = 5	0.8588	0.7907	0.4424	0.3731	0.5069	0.4167
Bagging	0.8618	0.8140	0.6134	0.3731	0.5764	0.3333
Random Forest	0.9324	0.9302	0.4944	0.3284	0.4653	0.2222
Adaboost	0.8059	0.8256	0.4572	0.3582	0.4931	0.4444
Gradient Boosting	0.7765	0.8023	0.5130	0.4627	0.5556	0.3889

Note that in total we had 426 data points of class "coup", 336 data points of class "attempted", and 181 data points of class "conspiracy".

Logistic Regression Model

We found the following predictors were least important (zeroed out by Lasso): **`1960s, 1970s, 1990s, 2000s, summer, winter, late, dissident, rebel, counter.`**

After bootstrapping, we had a better idea as to which predictors were most important. We note that the following coefficients did not include 0 in a 95% confidence interval after 100 bootstraps, sampling with replacement from our training data (753 data points): **`Africa, Europe, Oceania, South America, 1950s, 1960s, 1980s, dissident, rebel, palace, auto, resign, popular, and other`**. There is little overlap between these "non important" features from bootstrapping and the important features from Lasso regression, which confirms our list of important features.

Our baseline logistic regression model gave us a training accuracy of **0.6401** and a testing accuracy of **0.6032.**

Our logistic regression with Lasso penalty gave us a training accuracy of **0.6401** and a testing accuracy of **0.6032.**

Tree Models

In addition to calculating accuracy scores for our multi-class tree models, we also evaluated permutation importance to understand which predictors were the best. Similarly to the binary predictor models, all multi-class predictor trees found that **dissident** was the most important predictor. Following this, both boosting models found **palace** to be an important predictor and the random forest model also ranked **popular** and **foreign** as important predictors. Intuitively it makes sense that these predictors are important for the model since they indicate information about the characteristics of the coup. On the other hand, predictors such as **early**, **late**, **summer**, **fall**, and **other** were deemed unimportant. Once again these intuitively make sense since we would not expect the season or the time of the month to be especially important to the outcome of a coup.

Discussion & Conclusions

We found the baseline logistic regression model produced the best test score in both cases, which was unexpected to the team. It should be noted that in the case of the Logistic Lasso model,we found that the random state of the train-test split significantly impacted the accuracy of our model and which predictor coefficients went to 0. This suggests there is a fair degree of variance in the model and can be attributed to the size of the dataset, since we only had 942 observations. Outside of this, the performance ranking of each model differed between our two classification tasks - binary and multiclass. This helps underscore the importance of initially trying a broad range of models to find which one is best.

For Binary Classification – Realized or Unrealized Coup:

While the baseline logistic regression (no regularization) performed best, we **hypothesize that this is due to the randomization of the train-test-split**. Unlike in most occasions, the baseline logistic regression test score is actually higher than the train score, indicating that our random test split just happened to be very favorable to this model. The baseline logistic training accuracy is actually the lowest of all the models and it is the only model where the test accuracy is higher than the train. After the baseline logistic regression, the lasso-regularized logistic regression and random forest had the second highest accuracy scores on the test set. AdaBoost was third, followed by gradient boosting, and lastly the single decision tree. This, however, is expected. The single decision, even with hyperparameter tuning, cannot capture enough information alone to rival the ensemble methods. Bagging improves on its score by allowing us to build deeper individual trees without increasing the variance due to aggregation. Random forests then further improve upon the bagging model by decorrelating the trees. This gradual improvement in each model is reflected in the increase in test accuracy between each one. While the two boosting methods do not outperform the random forest model, they both easily outperform the single tree model, further solidifying our hypothesis that ensemble models could better learn decision boundaries. It should be noted that, excluding the poor single decision tree model, the difference between the best model (logistic regression without regularization) and the worst model (gradient boosting) is within 3.7% of each other. This close performance of the models indicates that with different random states, train-test-splits, and, most importantly, more data, the results may change.

For Multiclass Classification - Coup, Attempted, Conspiracy:

For multiclass prediction, the two logistic regression models, with and without regularization performed best (and equally well) on the test data. The gradient boosting model was second best (0.0001 lower score), followed by adaboost, random forest, the single decision tree, and the bagging model. Although the bagging model performed slightly worse than the single decision tree model on the test set (likely due to randomness with bootstrapping), all other ensemble tree models (random forest, adaboost, and gradient boosting) considerably outperformed the single decision tree. This reinforces the superiority of ensemble tree methods in comparison to single decision trees for more complex prediction problems. Unlike in the binary prediction, here both the boosting models outperform the random forest model and, also unlike in binary prediction, here gradient boosting outperforms adaboost.

Key Feature Selection from Regularization and Permutation Importance of Tree Models:

Easily the most important observation from looking at the coefficients with confidence intervals that do not contain 0 and permutation importance from the ensemble tree models is that the **feature 'dissident' is clearly seen as very important in both**. For our binary and multiclass tasks, the 95% CI interval for dissident coefficient values is far from 0, indicating that this feature is always used by the logistic regression models and is important to the performance of the model. In the random forest, AdaBoost, and gradient boosting model 'dissident' is by far and away the most important feature indicating that the accuracy score decreases significantly when this feature is not available to the model (replaced by random noise). This indicates that, for our models, **the 'dissident' coup type is most likely the strongest predictor of coup outcome**. While there are other predictors that are deemed significant by both Lasso regularization in logistic regression and permutation importance in ensemble tree models (such as the popular coup type), their confidence intervals or importance values are typically much closer to 0.

Strengths & Limitations:

Limitations Of Our Data and Models

After generating many models using different optimization techniques, we identified that:

- Our **baseline model produced the best test score** in both our binary and multi-class prediction models. We hypothesize that this is due to the train/test split of our limited amount of data.
- We had a **very small number of observations** to begin with. This led to high variation depending on how we train/test split our data, and is likely one main reason why it was difficult to improve the accuracy of our models. This combined with a **relatively very large feature space (due to one-hot-encoding)** made it difficult for our models to learn good decision boundaries.
- The **data in our model was unbalanced with respect to "event_type"**, the classes we were predicting in our multi-class model. In particular, we had 426 data points of class "coup", 336 data points of class "attempted", and 181 data points of class "conspiracy". As a result, our multi-class models had very different accuracies when grouped by class, which indicates that stratifying the data when creating the train/test split was insufficient and we should have oversampled/undersampled data from the classes to make the balance more equal.

Future Work:

- As noted in our Strengths & Limitations section, our data was not evenly distributed across the three "event_type" classes that our multi-class models were trying to predict. As can be seen in our Results section, this led to low accuracy scores when trying to predict "conspiracy" data points, since the "conspiracy" class was underrepresented. One area of future work would be to **oversample from this underrepresented class or undersample from the "coup" class**, which was overrepresented in our data set.
- Another area of future work would include **searching for a better, more extensive data set in order to answer our research questions**. We note that many of our predictors (such as the season and time of month) were not especially helpful in predicting the outcome of the coup. We were also limited by the size of our dataset, with less than 900 data points. More data points could help our models learn more about the dataset and generate a more generalizable model.

References

"Credit Scoring Using Logistic Regression and Decision Trees." MathWorks Documentation, https://www.mathworks.com/help/risk/creditscorecard-compare-logistic-regression-decision-trees.html.

Fuchs, Kirill. "Machine Learning: Classification Models." Medium, Fuzz, 17 Apr. 2017, https://medium.com/fuzz/machine-learning-classification-models-3040f71e2529.

Gong, Destin. "Top 6 Machine Learning Algorithms for Classification." Medium, Towards Data Science, 12 July 2022, https://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501.

Peyton, Buddy; Bajjalieh, Joseph; Shalmon, Dan; Martin, Michael; Bonaguro, Jonathan (2021): Cline Center Coup D'état Project Dataset. University of Illinois at Urbana-Champaign.https://doi.org/10.13012/B2IDB-9651987_V3
Ravi, Rakesh. "One-Hot Encoding Is Making Your Tree-Based Ensembles Worse, Here's Why?" Medium, Towards Data Science, 24 July 2022, https://towardsdatascience.com/one-hot-encoding-is-making-your-tree-based-ensembles-worse-heres-why-d64b282b5769.

Wijaya, Cornellius Yudha. "Categorical Feature Selection via Chi-Square." Medium, Towards Data Science, 12 Oct. 2021, https://towardsdatascience.com/categorical-feature-selection-via-chi-square-fc558b09de43.