Day 2: Classic Learners, Interpretation & Model Comparison

Machine Learning for Management Research

Istanbul Bilgi University · Python & Google Colab

Bruce A. Desmarais

Foundational learners for regression and classification

- **Support Vector Machines (SVM)**: *Optimal hyperplane classification in high-dimensional spaces.*
- **Trees & Random Forests**: *Efficient Nonlinear Representation in High Dimensions*
- **Nearest Neighbors (k-NN)**: *Leveraging similarity for classification.*

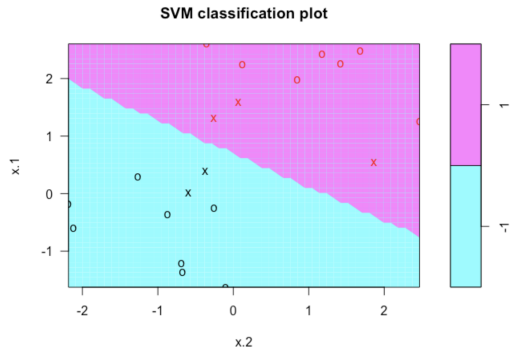
Support vector machines

Introduction to SVM Classifiers

- Classification/prediction based on feature/covariate regions.
- Developed for binary classification, extends to multi-class.
- Operates by finding the best margin (or "hyperplane") that separates classes.
- Handles nonlinearity using the kernel "trick".

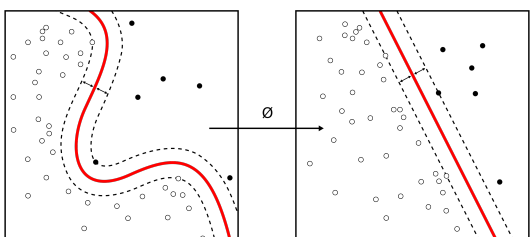
Maximizing the Margin

- SVM aims to find the widest possible separating margin.
- Data points touching the margin boundaries are termed as "support vectors".
- Goal: Maximize the margin while minimizing misclassification.



The Kernel Trick

- For non-linearly separable data, SVM can project data into a higher-dimensional space.
- Kernels (e.g., polynomial, radial basis function) allow such transformations.
- Data becomes linearly separable in the new space.



https://en.wikipedia.org/wiki/Support_vector_machine#/media/File:Kernel_Machine.svg

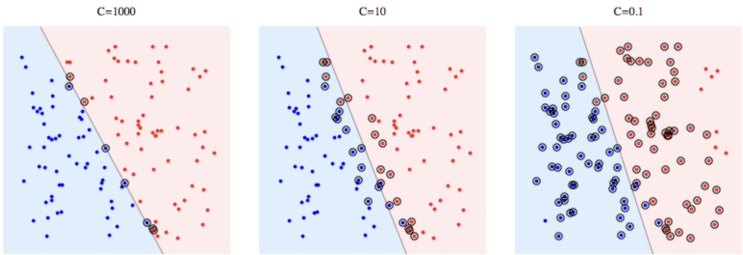
Overfitting in SVMs

- SVMs aim to find an optimal hyperplane that maximizes the margin between classes.
- Overfitting Example: An SVM that perfectly classifies all training points but forms a highly irregular boundary, performing poorly on new data.

Prevention of Overfitting in SVM

- Margin and Regularization: A wider margin can help in reducing overfitting.
- Regularization Parameter (C):
  - Smaller C: More misclassifications but better generalization.
  - Larger C: Fewer misclassifications but higher risk of overfitting.

Illustration of C parameter



A friendly introduction to Support Vector Machines (SVM).
https://towardsdatascience.com/a-friendly-introduction-to-support-vectormachines-svm-925b68c5a079

Key Hyperparameters in Support Vector Machines (SVM)

- **Kernel Type**: Common types include linear, polynomial, radial basis function (RBF), and sigmoid.
- **C (Regularization Parameter)**: A low value encourages a larger margin (simpler model) and a high value encourages classifying all training points correctly.
- **kernel-specific parameters**: Gamma for RBF Kernel, Degree for polynomial kernel, Coef0 for sigmoid kernel.

Advantages and Disadvantages

Advantages

- Effective in high-dimensional spaces.
- Memory efficient due to the use of support vectors.
- Versatile due to the kernel trick.

Disadvantages

- Highly computationally expensive.
- Sensitive to noisy data and outliers.
- Choice of the kernel and parameters can be complex.

Tree-based models

Introduction to Tree-based Classifiers

- Decision Trees: Hierarchical model of decisions and their consequences.
- Makes decisions based on asking a series of questions.
- Splits the data into subsets based on feature values.
- Can be visualized easily, intuitive for interpretation.

Basic tree structure

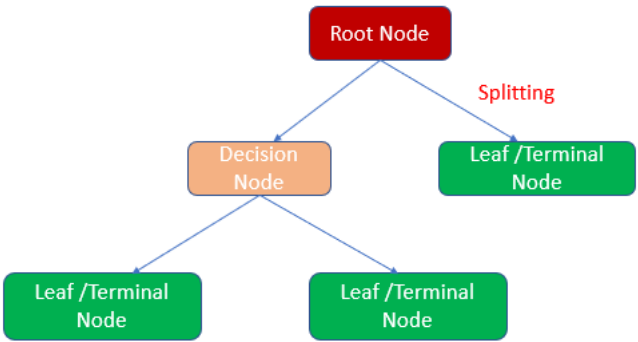
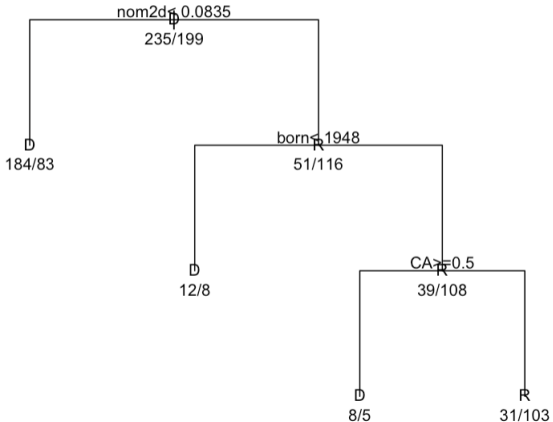


Image by Indhumathy Chelliah
medium.com/better-programming/understanding-decision-trees-in-machine-learning-86d750e0a38f

Example with data

116th U.S. House, predicting party



Overfitting example: Last split differentiates reps into CA group with 8/13 Democrats.

Recursive tree estimation

Global optimization of a tree is too complex in most datasets. With m binary predictors, there are 2^{2m} possible trees (e.g., 2^{1024} with 10 features).

Recursive partitioning is a *greedy* approximation method that is very fast, and follows a three-step process.

1. Split the root to optimally discriminate classes.
2. Split the branch that optimally discriminates, according to some fit metric.
3. Repeat step (2) until a stopping criterion is satisfied.

- Splits are sequential, but gains can be calculated in parallel.
- Gini index for splits: make leafs more lopsided

Tree hyperparameters

Use cross-validation to tune stopping criteria

- Minimum cases in a node before splitting
- Max depth
- Minimum split performance
- Minimum cases in leaf

Two modes of tree-building

- Top-down: stop when stopping criteria triggered
- bottom-up: build full/pure tree, then prune based on split performance

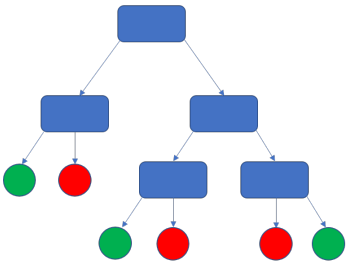
Random forests

- Single trees overfit data due to in-sample flexibility.
- Interpretable and fast, but poor out-of-sample performance.
- Flexibility can be retained, while improving robustness and predictive performance, through ensembles of trees.
- Random Forests are an ensemble learning method.
- Combines multiple decision trees to improve accuracy and control over-fitting.

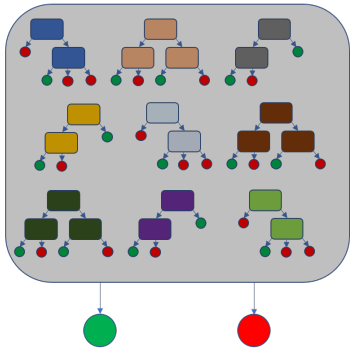
How Random Forests Work

- **Bootstrap Aggregating (Bagging)**: Random subsets of the training dataset are created.
- **Decision Tree Creation**: A decision tree is grown on each subset.
- **Random Feature Selection**: At each split in the tree, a random subset of features is considered.
- **Aggregation**: Final prediction is made by averaging (regression) or majority voting (classification) across all trees.

Decision Trees vs. Random Forest



Decision Tree



Random Forest

Key Features of Random Forests

- **Reduction in Overfitting**: Due to averaging/majority voting across multiple trees.
- **Handling Missing Values**: Capable of handling missing data in training.
- **Feature Importance**: Provides insights into the importance of each feature in prediction.
- **Versatility**: Effective for both categorical and continuous input and output variables.

Advantages of Random Forests

- **High Accuracy**: Offers one of the most accurate learning algorithms.
- **Robust to Noise**: Can handle noise and outliers in the dataset.
- **Ease of Use**: Generally requires very few hyperparameters tuning.

Random forest hyperparameters

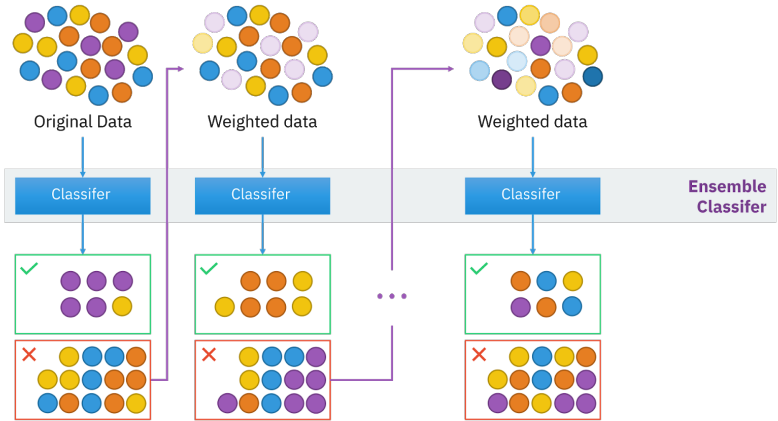
The phrase "random forest" typically only refers to bagging, but the following hyperparameters also apply to boosting and stacking of trees.

- Number of trees in the forest
- Number of features sampled
- Minimum cases in a leaf
- Maximum number of leaves

Introduction to XGBoost

- XGBoost stands for eXtreme Gradient Boosting.
- It is an advanced implementation of gradient boosting algorithm.
- Known for its speed and performance.
- Widely used in machine learning competitions and industry applications.

XGBoost Illustration



XGBoost: Everything You Need to Know [Online]. Available on: neptune.ai

Key Features of XGBoost

- **Performance**: Highly efficient, scalable, and fast.
- **Regularization**: Includes L1 (Lasso Regression) and L2 (Ridge Regression) regularization to prevent overfitting.
- **Handling Missing Values**: Automatically handles missing data.
- **Tree Pruning**: Uses depth-first approach and shaves off unnecessary branches, improving efficiency.

How XGBoost Works

- Builds multiple decision trees sequentially.
- Each tree corrects the errors of the previous ones.
- Uses gradient descent to minimize loss when adding new models.
- Objective function = Training Loss + Regularization Term.

Nearest neighbor classification

Introduction to Nearest Neighbor Classifiers

- A non-parametric, lazy learning algorithm.
- Predicts based on the similarity between instances.
- Commonly used in pattern recognition.
- Example: If you have similar political views as your neighbors, you may vote similarly.

Working Principle

- For a new instance, compute distance to all training samples.
- Consider the closest k instances (neighbors).
- For classification: Take a majority vote.
- For regression: Compute a mean or median.

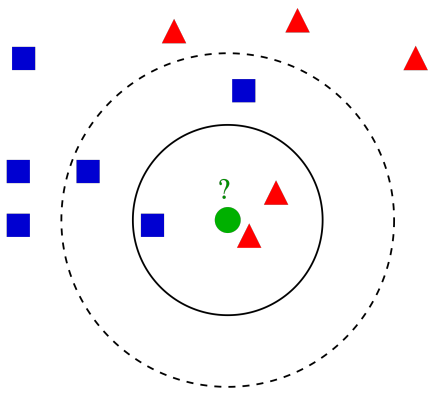


Image from https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm/media/File : KnnClassification.svg

Distance Metrics

- Euclidean Distance: $d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$
- Manhattan Distance: $d(x, y) = \sum_i |x_i - y_i|$
- Minkowski Distance: $d(x, y) = (\sum_i |x_i - y_i|^p)^{1/p}$
- Importance of feature scaling in distance-based algorithms.

Advantages and Disadvantages

Advantages

- Simple and intuitive.
- No assumptions about data distribution.
- Useful for nonlinear data.

Disadvantages

- Computationally intensive.
- No concise function for classification.
- Sensitive to irrelevant features.

EEG readings of student engagement

Figure 2. Examples of dry-wireless EEG systems. (a) MindWave 2 (NeuroSky); (b) Muse (InteraXon Inc.); (c) ENOBIO 8 (Neuroelectrics); (d) BR8 (BRI); (e) EPOC (Emotiv); (f) Quick (Cognionics)

Huang, H.W., King, J.T. and Lee, C.L., 2020. The New Science of Learning. Educational Technology Society, 23(4), pp.1-13.

Key Hyperparameters in Nearest Neighbor Classification

- **Number of Neighbors (k)**: A small value of k means that noise will have a higher influence on the result, while a large value makes it computationally expensive.
- **Distance Metric**: Common metrics include Euclidean, Manhattan, Minkowski, and Hamming distance.
- **Weights**: Options include uniform weights (all neighbors have equal weight) or distance-based weights (closer neighbors have more influence).

Multi-learner comparison

Common setup for ML predictive modeling

- Data is split into training and test sets.
- Training set is used to train the models.
- Test set is used to evaluate the performance of the models.
- Common splits: 70-30, 80-20, or using cross-validation.

Hyperparameter Selection

- Methods of selection: Grid Search, Random Search, Bayesian Optimization.
- Importance of selecting the right hyperparameters for model optimization.
- Example hyperparameters for each algorithm (SVM - C, kernel; KNN - k; Random Forests - number of trees; XGBoost - learning rate, depth).

Model Training

- Training models on the training dataset.
- Monitoring for convergence and overfitting.
- Adjusting hyperparameters based on training performance.

Model Comparison Using Test Data

- Evaluating trained models on the test dataset.
- Comparing performance metrics: accuracy, precision, recall, F1-score, etc.
- Analyzing results to determine the best-performing model.
- Consider combinations of models

Predicting school dropout

TABLE 8. Area under the ROC Curve for Models Predicting School Dropout

Machine learning algorithm	*School dropout measures*		
	Dropout in t + 1 or t + 2	*Dropout in t + 1*	*Dropout in t + 2*
glmnet	0.866	0.893	0.857
GAM	0.865	0.892	0.854
GBM	0.863	0.891	0.851
Lasso	0.858	0.885	0.845
SVM	0.853	0.843	0.803
RF	0.849	0.875	0.844

Source: Author's calculations, using administrative data sets from the Chilean Ministry of Education and Ministry of Social Development.
Notes: The area under the ROC curve measures the overall predictive performance of the model. A machine learning algorithm that makes no predictive mistakes has an AUC of 1.0, while a model that predicts at random should achieve an AUC near 0.5.

CRESPO, C. (2020). Two Become One: Improving the Targeting of Conditional Cash Transfers with a Predictive Model of School Dropout. Economía, 21(1), 1–46.

Predicting grocery sales

TABLE 1—MODEL COMPARISON: PREDICTION ERROR

	Validation		Out-of-sample		Percent
	RMSE	SE	RMSE	SE	weight
Linear	1.169	0.022	1.193	0.020	6.62
Stepwise	0.983	0.012	1.004	0.011	12.13
Forward stagewise	0.988	0.013	1.003	0.012	0.00
LASSO	1.178	0.017	1.222	0.012	0.00
Random forest	0.943	0.017	0.965	0.015	65.56
SVM	1.046	0.024	1.068	0.018	15.69
Bagging	1.355	0.030	1.321	0.025	0.00
Logit	1.190	0.020	1.234	0.018	0.00
Combined	0.924		0.946		100.00

Bajari, P., Nekipelov, D., Ryan, S.P. and Yang, M., 2015. Machine learning methods for demand estimation. American Economic Review, 105(5), pp.481-485.

Take away

- Classic learners represent backbone of predictive machine learning.
- Best approach is typically to try out two or more families, and see what works bests.
- Interpretability remains a major limitation in application.

Permutation-based Feature Importance

Introduction to Feature (variable) Importance

- Feature importance measures the significance of individual features in contributing to a machine learning model's predictive power.
- Understanding feature importance helps in model interpretation, simplification, and provides insights into the data's underlying structure.
- Permutation-based feature importance is a model-agnostic method that evaluates the impact of shuffling individual features on model performance.

Methodology of Permutation-Based Feature Importance

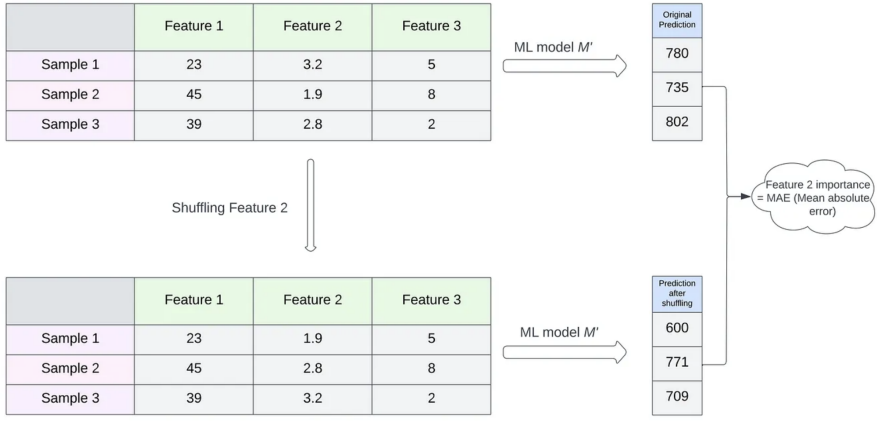
1. Train the model using the entire dataset.

2. Evaluate the model's performance using a chosen metric (e.g., accuracy for classification, MSE for regression).

3. For each feature/variable:
   - Randomly shuffle the values of that feature across all samples.
   - Re-evaluate performance on this perturbed dataset.
   - Calculate the importance score as the change in performance.

4. Rank the features based on their importance scores.

Quantifies how much each feature contributes to predictive performance.



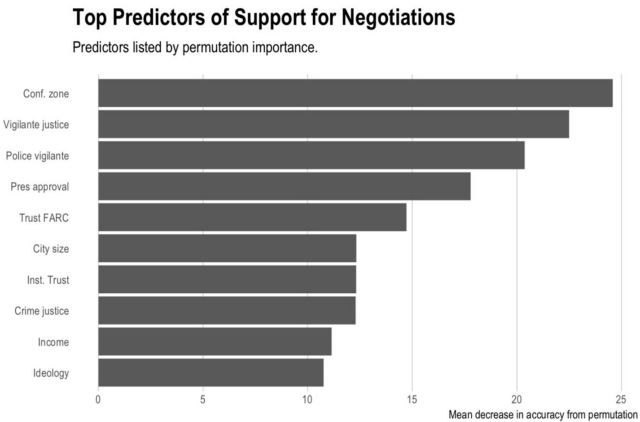
https://medium.com/@gauravagarwal_14599/explainable-ai-xai-permutation-feature-importance-7be0da6f0600

Advantages

- Model Agnostic: Applicable to any machine learning model.
- Easy to Implement and Interpret: Does not require complex mathematical formulations.
- Provides insights into the robustness of the model against feature noise.

Limitations

- Less informative for features with high inter-correlations.
- Computationally expensive for models with a large number of features.
- Does not capture causal relationships, only correlations.



Montoya, A.M. and Tellez, J., 2020. Who Wants Peace? Predicting Civilian Preferences in Conflict Negotiations. Journal of Politics in Latin America, 12(3), pp.252-276.

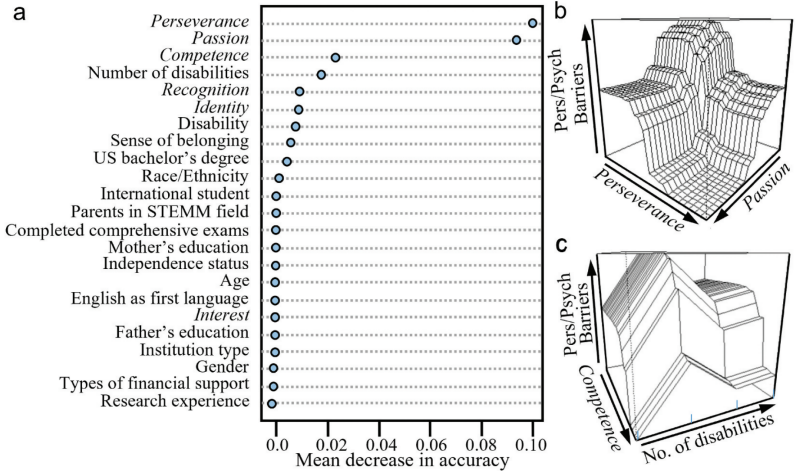


Figure 2. Full cRF model for *personal/psychological barriers* (Pers/Psych). (A) Permutation importance plot for the model including all variables. (B) Bivariate partial dependence plot of the grit subscales, *perseverance* and *passion*, in predicting Pers-B. (C) Bivariate partial dependence plot of the number of disabilities with which participants identified and the science identity subscale, *competence*, in predicting *personal/psychological barriers*.

Sanchez, K.A., Bevan, A.J., Vita, A.A., Royse, E.A., Januszkiewicz, E. and Holt, E.A., 2023. Grit matters to biology doctoral students' perception of barriers to degree completion. Journal of Biological Education, pp.1-24.

Interpretation & model comparison

Why interpret a "black box"?

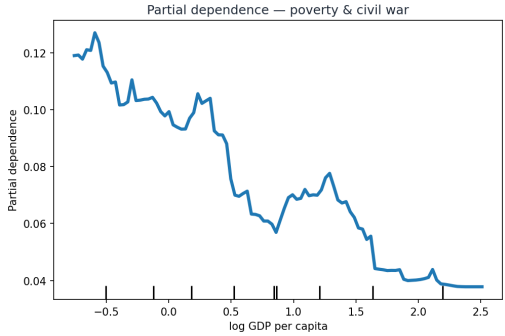
- Flexible learners predict well but hide *why*.
- For research we still need to connect predictions back to substantive variables.
- Three model-agnostic tools: **permutation importance**, **partial dependence**, **SHAP**.

Permutation importance

- Shuffle one feature; measure how much test performance drops.
- Large drop $\Rightarrow$ the model relied on that feature.
- Model-agnostic, but can be misleading when features are correlated.

Partial dependence plots (PDP)

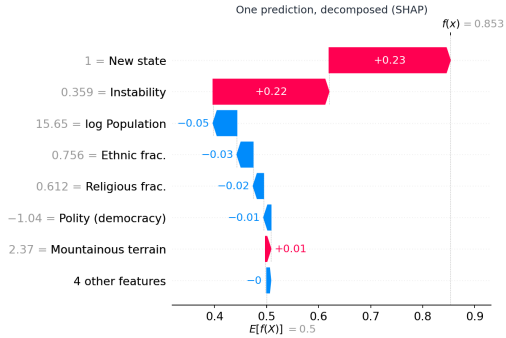
- *Holding everything else fixed, how does the prediction change as one feature varies?*
- Recipe: sweep the feature across its range; at each value set it for *every* observation and average the model's predictions.
- The curve is the feature's average marginal effect — a *shape*, not one coefficient (it can bend, plateau, or reverse).
- In Python: `sklearn.inspection.PartialDependenceDisplay`.
- Caveat: with correlated features it averages over unrealistic combinations.



Civil-war onset (Fearon–Laitin): predicted risk falls as GDP per capita rises.

SHAP values

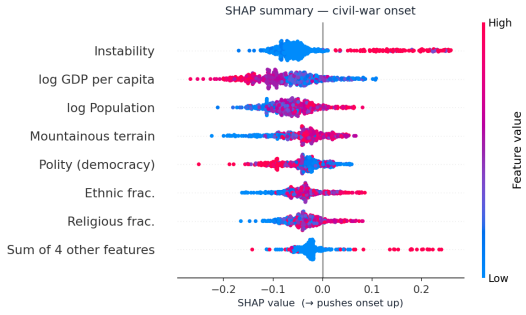
- SHAP explains a *single* prediction as a sum of feature contributions: prediction = base value + $\sum_j \phi_j$.
- The ϕ_j are **Shapley values** — each feature's average marginal contribution over all orderings; the unique consistent, locally-accurate attribution.
- **Local**: a waterfall / force plot explains one case. **Global**: stack many into a **beeswarm**.
- In Python: the `shap` library; TreeSHAP is a fast exact method for tree ensembles.



One country-year: base rate + "new state" + "instability" ⇒ high predicted onset.

Reading a SHAP summary (beeswarm)

- Features ordered top-to-bottom by overall importance (mean $|\phi_j|$).
- Each dot is one observation; it spreads left/right by its SHAP value: right = pushes the prediction *up*, left = *down*.
- Color = the feature's *value* (red high, blue low), so you read direction: "high $X \Rightarrow$ higher outcome."
- A clean colour split = a strong monotone effect; mixed colours hint at interactions.



Comparing models fairly

Model	CV metric	Notes
Logistic / OLS	baseline	interpretable, fast
Random forest	often better	captures interactions
Gradient boosting	often best	needs tuning

- Identical folds, one metric (ROC-/PR-AUC or RMSE), report uncertainty.
- Tune hyperparameters *inside* the training folds — never on the test set.

A worked result (turnout)

- Reproduced OLS of state turnout (Kuhlmann & Lewis 2017): in-sample $R^2 = 0.72$.
- 10-fold CV: gradient boosting cuts RMSE **23%** below OLS (a GAM alone, 15%).
- Lesson: when the signal is nonlinear/interactive, the learner predicts better — and SHAP/partial dependence show *how*.
- Notebook: `notebooks/03_variable_selection.ipynb`, `06_interpretation.ipynb`.

ML in the research pipeline

1. Question first: prediction vs. explanation sets the model and metric.
2. Honest evaluation: cross-validate; never tune on the test set.
3. Compare fairly: same folds, same metric, report uncertainty.
4. Interpret: SHAP / partial dependence connect the model to substance.
5. Document & share: code + data so others can reproduce.

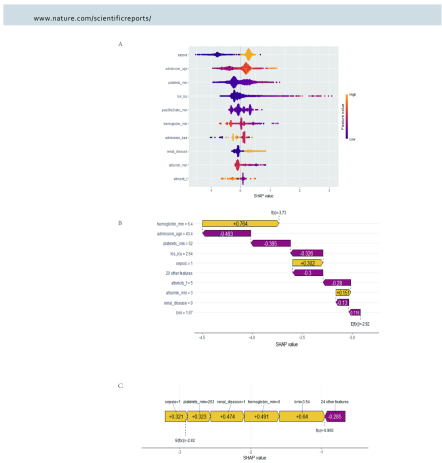
Course wrap-up

- You now have the full toolkit: evaluation, regression extensions, regularization, trees/boosting, SVM/k-NN, interpretation.
- The discipline that matters most: **honest out-of-sample evaluation**.
- Reproduce, extend, compare, interpret — then make the substantive claim.
- All code is in the Colab notebooks; bring your own data and reuse the pipelines.

Further reading

Five studies that extend today's examples — with a figure from each

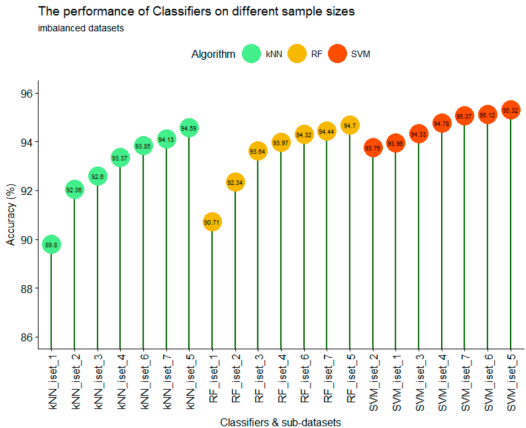
XGBoost + SHAP in intensive care



An ICU application of **XGBoost** with **SHAP**. The model predicts pressure injuries in ventilated patients (MIMIC-IV); the SHAP summary ranks the drivers and shows the *direction* of each feature's effect — the boosting-plus-interpretation pipeline from today.

Zheng, L., Xue, Y., Yuan, Z. & Xing, X. (2025). Explainable SHAP-XGBoost models for pressure injuries among ICU patients on mechanical ventilation. *Scientific Reports* 15, s41598-025-92848-2.

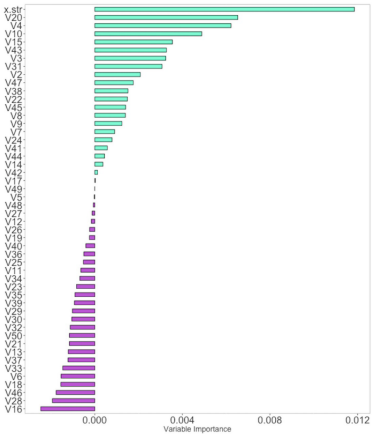
SVM vs. k-NN vs. random forest



A head-to-head of the three classic learners from today on satellite land-cover mapping (Sentinel-2). Accuracy is compared across training-set sizes: SVM is most accurate and least sensitive to sample size, then RF, then k-NN — the "compare a few families" lesson, made concrete.

Thanh Noi, P. & Kappas, M. (2018). Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery. *Sensors* 18(1), 18.

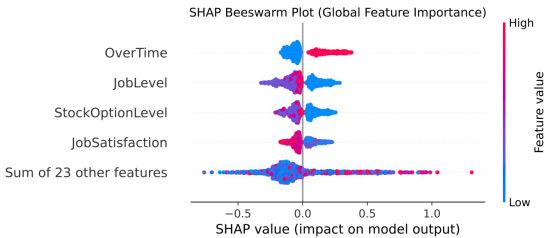
Random forests in conflict research



A political-science application of **random (survival) forests**. Predicting the durability of post-conflict peace, the variable-importance plot ranks which factors — power sharing, third-party mediation — most shape the forecast, then partial-dependence plots show how.

Whetten, A. B., Stevens, J. R. & Cann, D. (2021). The implementation of random survival forests in conflict management data. *PLOS ONE* 16(5).

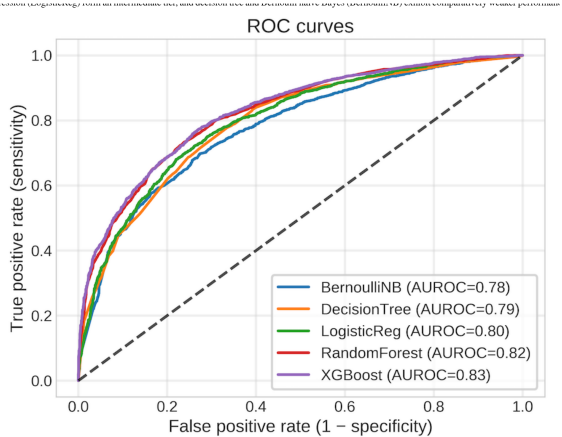
Explainable AI for employee attrition



A management/HR application. Tree ensembles predict which employees will leave; the **SHAP** beeswarm shows overtime, job level, and stock options as the strongest drivers, and *how* each pushes the risk up or down — interpretation for a business decision.

AL-Ali, M., Alwateer, M., Alsaedi, S. A., Balaha, H. M., Badawy, M. & Elhosseini, M. A. (2026). Integrating machine learning and explainable AI for employee attrition prediction in HR analytics. *Scientific Reports* 16, s41598-026-36424-2.

Comparing learners fairly: cancer mortality



A worked *model-comparison*. Five learners (naïve Bayes, tree, logistic, random forest, XGBoost) are trained to predict metastatic-cancer mortality and compared on one ROC axis. XGBoost and the random forest edge out logistic regression — the honest bake-off from the comparison section.

Nalela, P., Rao, D. & Rao, P. (2026). Explainable AI for predicting mortality risk in metastatic cancer. *JMIR Cancer* 12, e74196.

References

Works cited (1/2)

- AL-Ali, M., Alwateer, M., Alsaedi, S. A., Balaha, H. M., Badawy, M. & Elhosseini, M. A. (2026). Integrating machine learning and explainable AI for employee attrition prediction in HR analytics. *Scientific Reports* 16, s41598-026-36424-2.
- Bajari, P., Nekipelov, D., Ryan, S. P. & Yang, M. (2015). Machine learning methods for demand estimation. *American Economic Review* 105(5), 481–485.
- Crespo, C. (2020). Two become one: Improving the targeting of conditional cash transfers with a predictive model of school dropout. *Economía* 21(1), 1–46.
- Huang, H. W., King, J. T. & Lee, C. L. (2020). The new science of learning. *Educational Technology & Society* 23(4), 1–13.
- Kuhlmann, R. & Lewis, D. C. (2017). Legislative term limits and voter turnout. *State Politics & Policy Quarterly*.
- Montoya, A. M. & Tellez, J. (2020). Who wants peace? Predicting civilian preferences in conflict negotiations. *Journal of Politics in Latin America* 12(3), 252–276.

Works cited (2/2)

- Nalela, P., Rao, D. & Rao, P. (2026). Explainable AI for predicting mortality risk in metastatic cancer. *JMIR Cancer* 12, e74196.
- Sanchez, K. A., Bevan, A. J., Vita, A. A., Royse, E. A., Januszkiewicz, E. & Holt, E. A. (2023). Grit matters to biology doctoral students' perception of barriers to degree completion. *Journal of Biological Education*.
- Thanh Noi, P. & Kappas, M. (2018). Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery. *Sensors* 18(1), 18.
- Whetten, A. B., Stevens, J. R. & Cann, D. (2021). The implementation of random survival forests in conflict management data. *PLOS ONE* 16(5).
- Zheng, L., Xue, Y., Yuan, Z. & Xing, X. (2025). Explainable SHAP-XGBoost models for pressure injuries among ICU patients on mechanical ventilation. *Scientific Reports* 15, s41598-025-92848-2.