Project Research Laboratory

Extending Boruta
- a popular feature selection ML algorithm -

By Clémence Mottez

Supervised by Thomas SIMONSON and Ivan REVEGUK

Feature selection

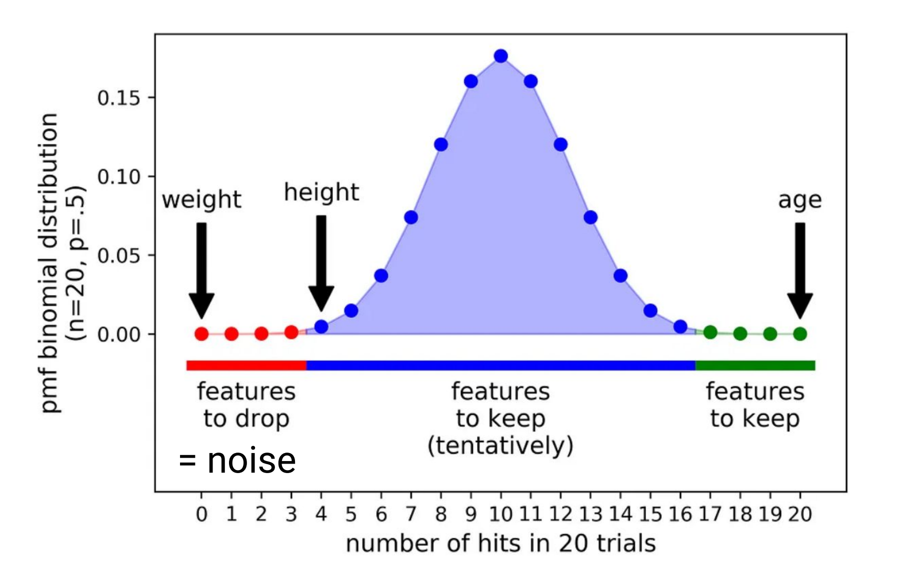
- Boruta = feature selection algo

- Why important in ML?
  - Reduce size data
    - Slow down algo
  - Remove unnecessary / redundant features
    - Reduce noise
    - Improve model accuracy
  - Improve interpretability

Boruta developed by Kursa and Rudnicki, 2010

- Why Boruta?
  - Shadow features
    - Eliminate correlation
  - Random forest
  - > Threshold = highest shadow
    = “hit”
  - Random values -> several runs
  - Handle correlated features
    missing values

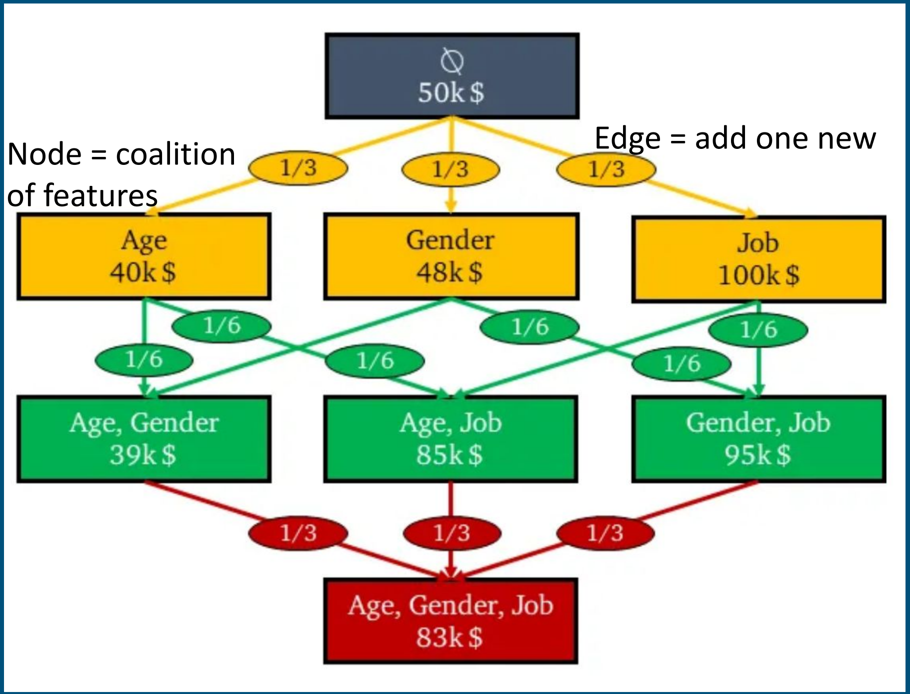
Area refusal | Irresolution | Acceptance



Binomial distribution and positioning of the features

SHAP SHapley Additive exPlanations

- Contribution each feature to prediction
- Train model on each node + predict x0
    - 2 ^ F nodes
- Compute marginal contribution of each feature to the prediction x0



-> Help to gain insights into importance of features at each step

-> Selects subset that maximize performance

-> Uses to rank most important features

$\text{SHAP\_Age}(x_0) = \text{-}11.33\text{k } \$$

$\text{SHAP\_Gender}(x_0) = \text{-}2.33\text{k } \$$

Strongest positive influence on the predicted outcome -> $\text{SHAP\_Job}(x_0) = +46.66\text{k } \$$

eBoruta

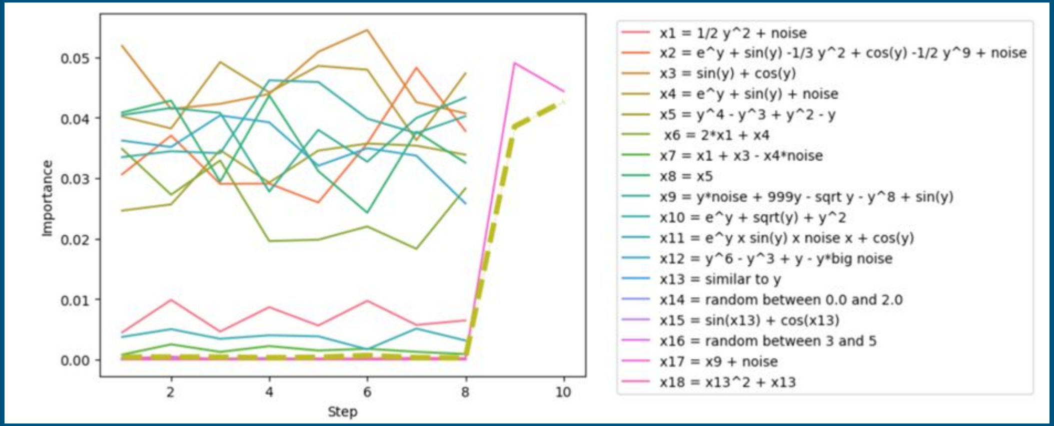
- Add SHAP importance in Boruta
  - Model-agnostic (not uniquely Random Forest)
- Add parameters
  - P-value, percentile, test size, …
  - Guide selection
  - More flexibility in selection process

Dataset

- Create my own
  - Control over columns
  - Normal outcome
- "y" : random integer 0-5
- "x" : + - correlated with "y”

- Didn’t manage to mislead the algo

X1	$\frac{1}{2}y^2 + Noise$
X2	$e^y + sin(y) - \frac{1}{3}y^2 + cos(y) - \frac{1}{2}y^9 + Noise$
X3	$5y^4 - \frac{1}{2}y^3 + y^4 - \frac{1}{5}y^2 + y + Noise$
...	...
X17	$Random\ number\ between\ 0\ and\ 5 + noise$
X18	$Noise$



Importance plot

Data

- Confirms the effectiveness of Boruta
- Can’t analyse algo with a "perfect" dataset
  + more interesting real world data
- Price of oil
  - Parameters that could influence price
  - Others with weaker correlation
  - Asked domain experts

Feature	Importance
x3 = sin(y) + cos(y)	0.048538
x4 = e^y + sin(y) + noise	0.043161
x10 = e^y + sqrt(y) + y^2	0.041615
x9 = y*noise + 999y - sqrt y - y^8 + sin(y)	0.039067
x5 = y^4 - y^3 + y^2 - y	0.034180
x8 = x5	0.034147
x2 = e^y + sin(y) -1/3 y^2 + cos(y) -1/2 y^9...	0.031540
x12 = y^6 - y^3 + y - y*big noise	0.030831
x6 = 2*x1 + x4	0.022180
x1 = 1/2 y^2 + noise	0.010080
x11 = e^y x sin(y) x noise x + cos(y)	0.004636
x7 = x1 + x3 - x4*noise	0.001747
shadow_x17 = x9 + noise	0.000431
shadow_x7 = x1 + x3 - x4*noise	0.000405
shadow_x14 = random between 0.0 and 2.0	0.000342
shadow_x1 = 1/2 y^2 + noise	0.000324
x14 = random between 0.0 and 2.0	0.000230
shadow_x9 = y*noise + 999y - sqrt y - y^8 + si...	0.000223
shadow_x4 = e^y + sin(y) + noise	0.000203
x17 = x9 + noise	0.000198
shadow_ x6 = 2*x1 + x4	0.000145
shadow_x2 = e^y + sin(y) -1/3 y^2 + cos(y) -...	0.000112
shadow_x13 = similar to y	0.000079
shadow_x11 = e^y x sin(y) x noise x + cos(y)	0.000078
shadow_x12 = y^6 - y^3 + y - y*big noise	0.000060
x18 = x13^2 + x13	0.000049
x16 = random between 3 and 5	0.000037
shadow_x15 = sin(x13) + cos(x13)	0.000032
x15 = sin(x13) + cos(x13)	0.000031
shadow_x3 = sin(y) + cos(y)	0.000018
x13 = similar to y	0.000014
shadow_x5 = y^4 - y^3 + y^2 - y	0.000009
shadow_x18 = x13^2 + x13	0.000008
shadow_x16 = random between 3 and 5	0.000008
shadow_x8 = x5	0.000000
shadow_x10 = e^y + sqrt(y) + y^2	0.000000

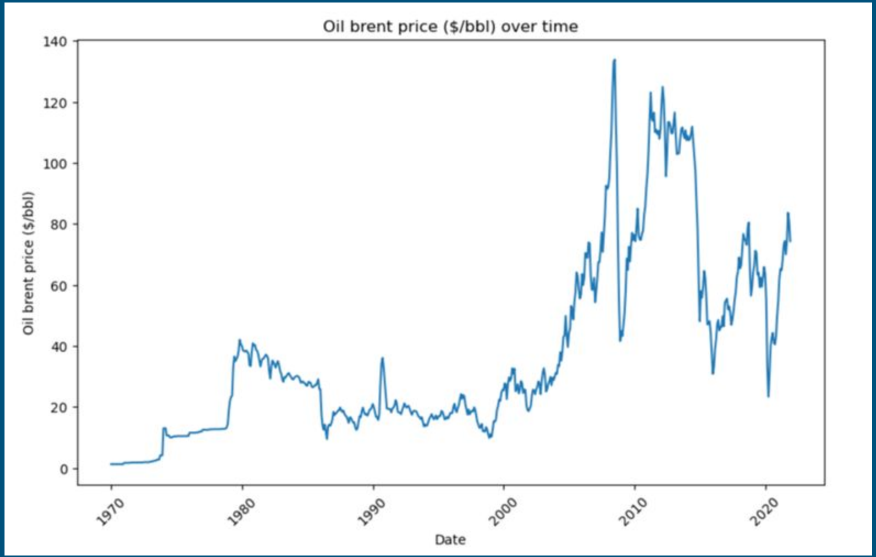
Data

- Normalization

$$x = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- Analysis

- My dataset
  - Monthly data from 1970 to 2022
  - = 624 rows x 23 columns

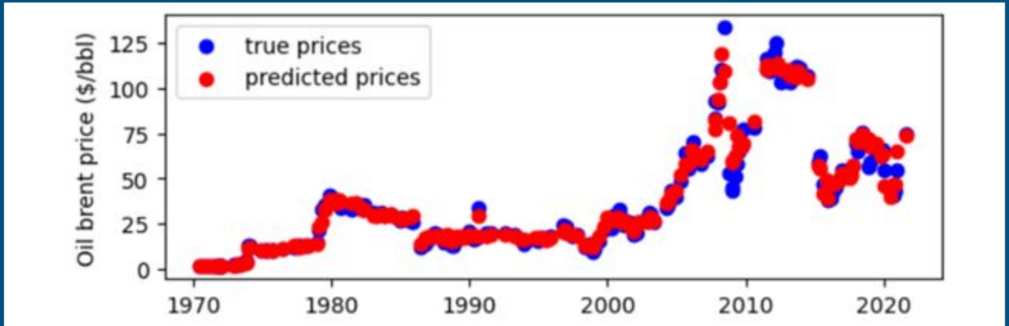


Model

	XGB Regressor	Decision Tree	Random Forest
MSE	**23**	40	29

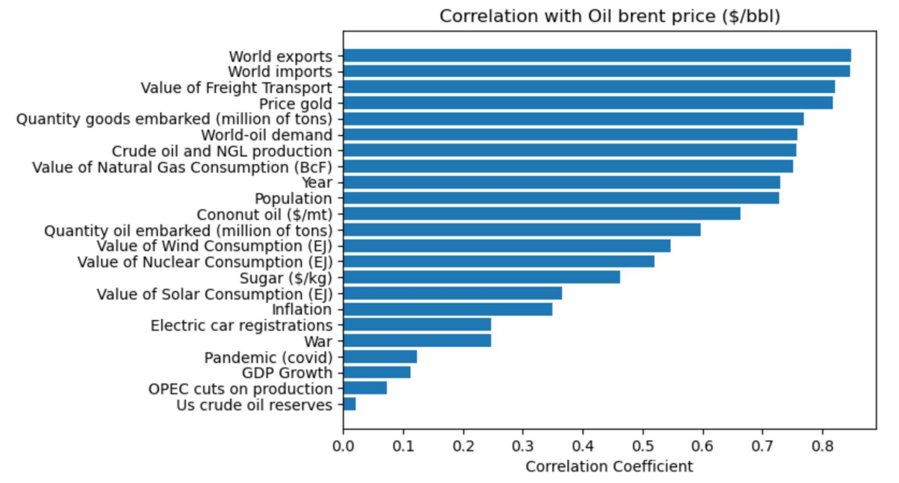
- Needed for feature selection
  - Cross-validation
    - Random Forest, Decision Tree, Polynomial Regression
  - Hyperparameter tuning
  - Maximize R2 score and minimize MSE

XGB regressor



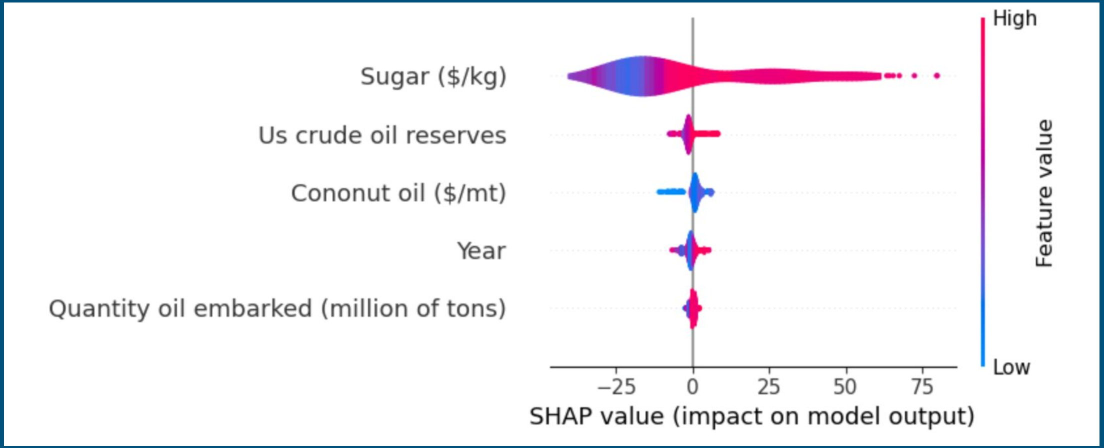
Filter models

- Computationally efficient
- Involve statistical measures such as correlation



Embedded models

- Incorporates feature selection within the model training process
  - Optimizes both model performance and feature selection -> select features based on their contribution to the accuracy

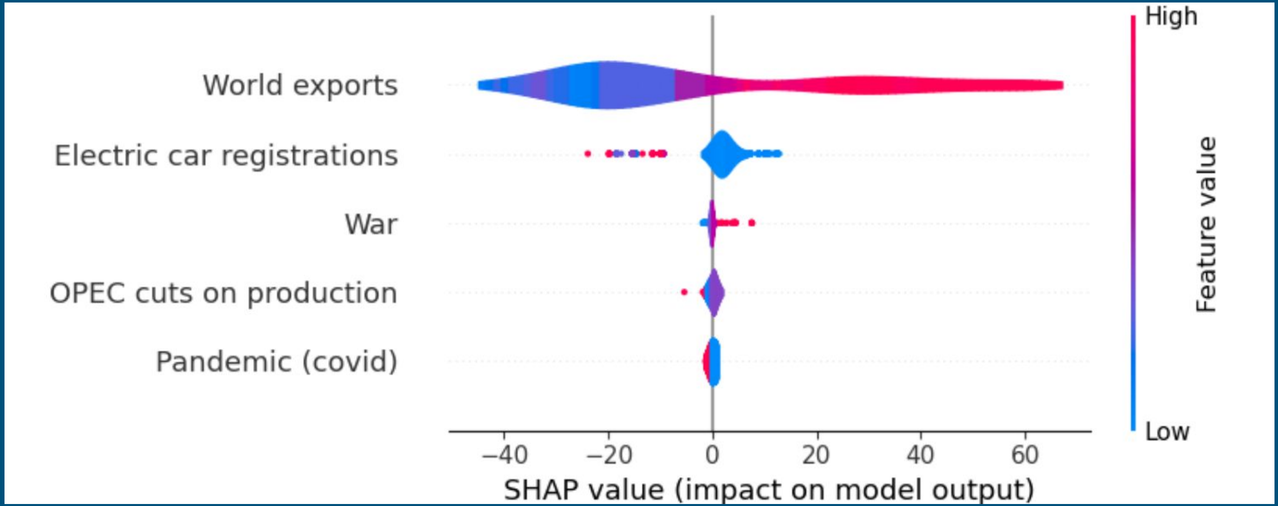


Wrapper models

- eBoruta
- Iteratively select and evaluate different subsets of features to find the subset that yields the best performance
    - Computationally expensive but accurate results

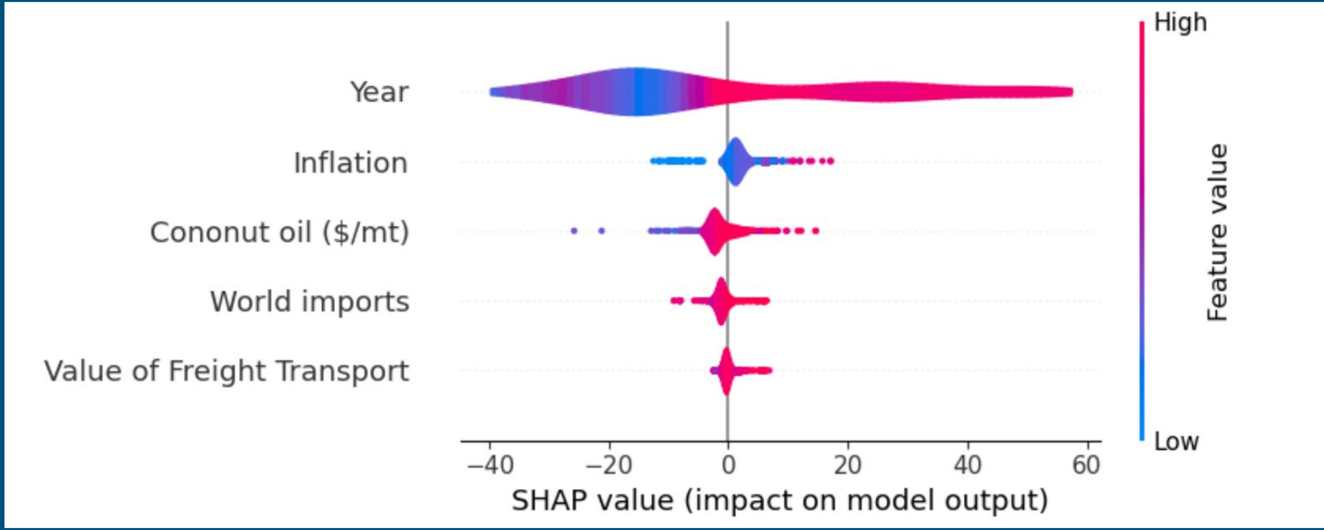
Wrapper models

Sequential Forward Selection



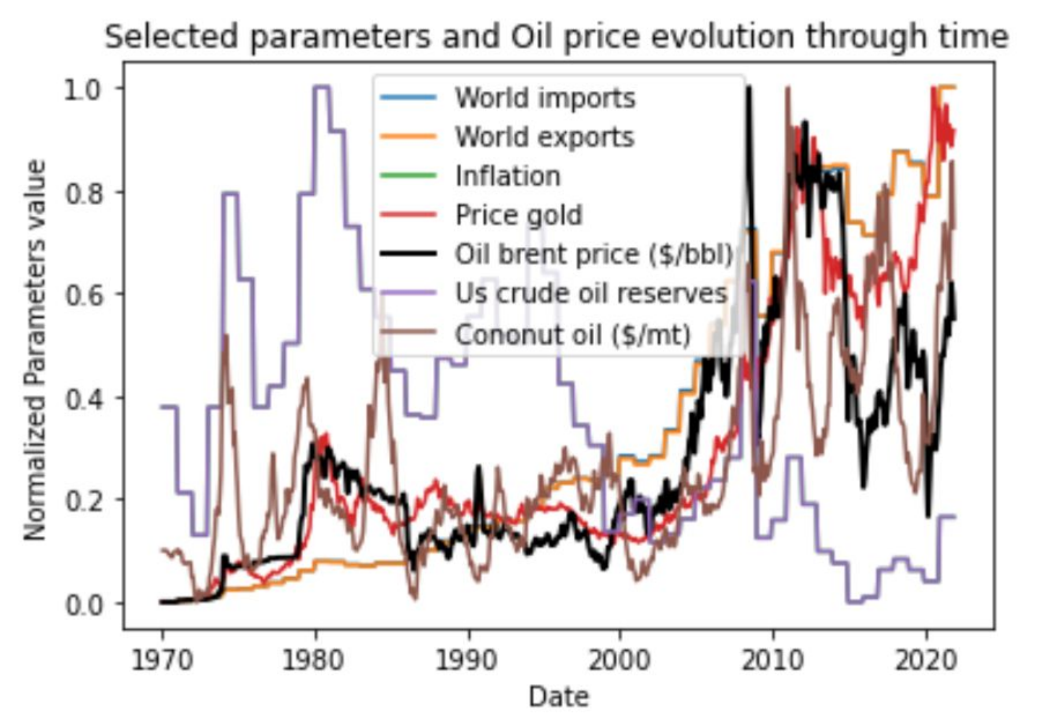
Wrapper models

Recursive Feature Elimination



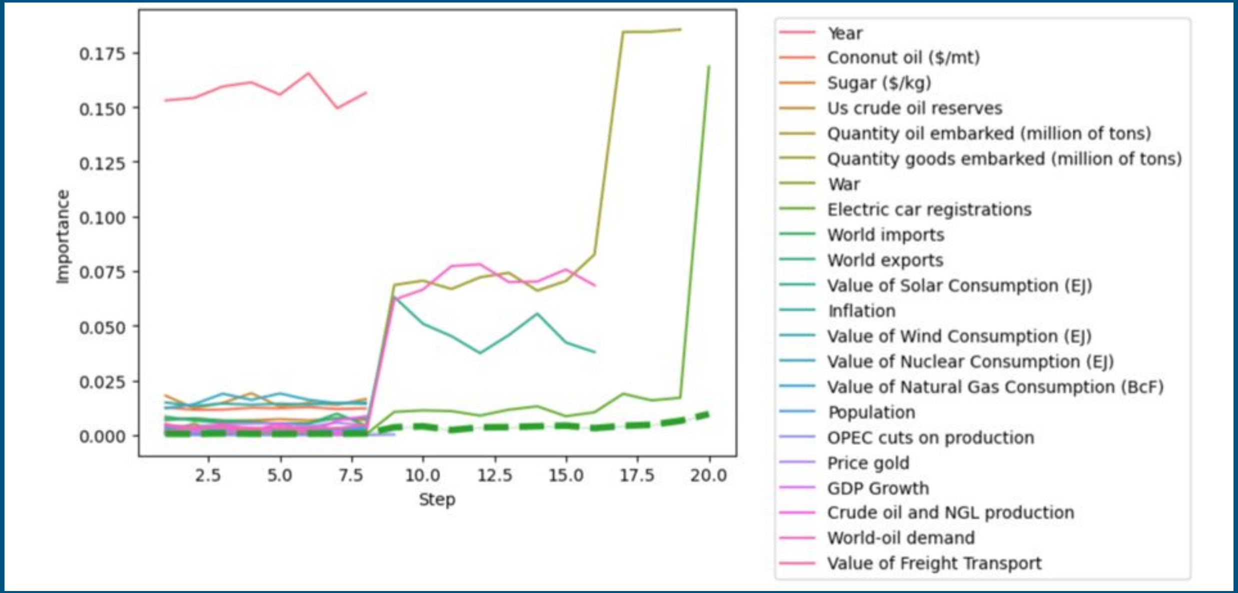
Selection

- Year
- World imports
- World exports
- Inflation
- Price of Gold
- Price coconut oil
- US crude oil reserve



eBoruta

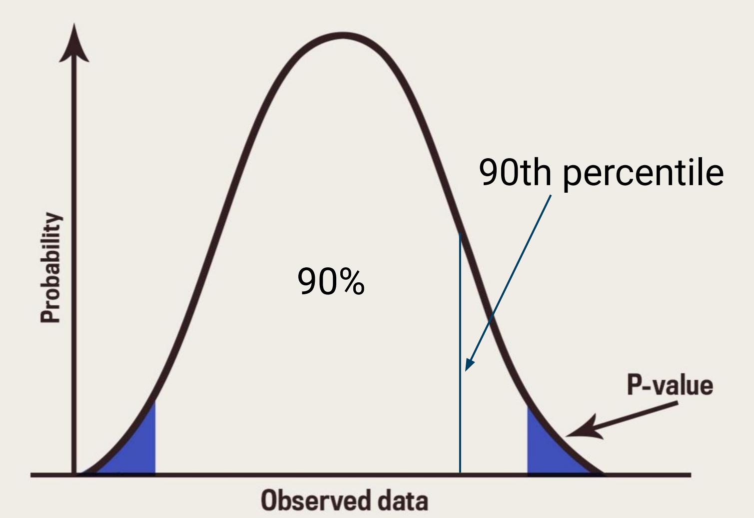
- Default parameters
- Repeat to ensure stability



Feature	Importance
Year	0.159842
Inflation	0.015150
Value of Nuclear Consumption (EJ)	0.014833
Cononut oil ($/mt)	0.012274
Us crude oil reserves	0.012178
Sugar ($/kg)	0.008447
World imports	0.006078
Value of Freight Transport	0.005856
World exports	0.005784
Price gold	0.005459
World-oil demand	0.003815
Value of Wind Consumption (EJ)	0.003145
Population	0.003127
Value of Natural Gas Consumption (BcF)	0.002677
Crude oil and NGL production	0.002076
War	0.001907
GDP Growth	0.001743
Value of Solar Consumption (EJ)	0.001622
Quantity goods embarked (million of tons)	0.001508
Quantity oil embarked (million of tons)	0.000988
Electric car registrations	0.000724
OPEC cuts on production	0.000092

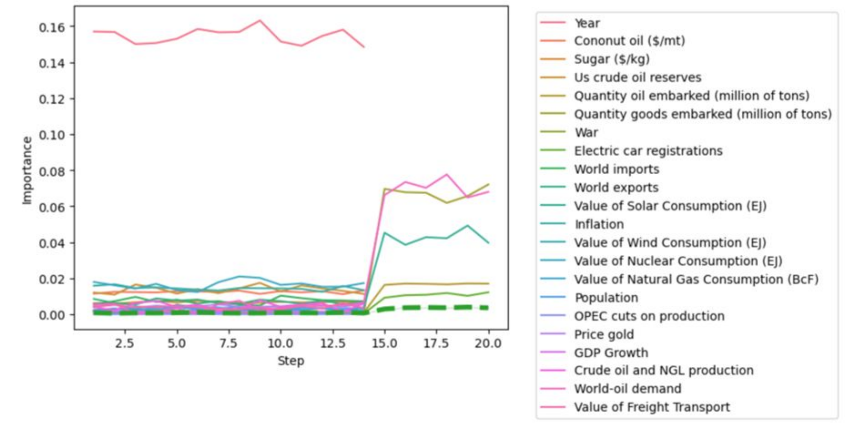
eBoruta

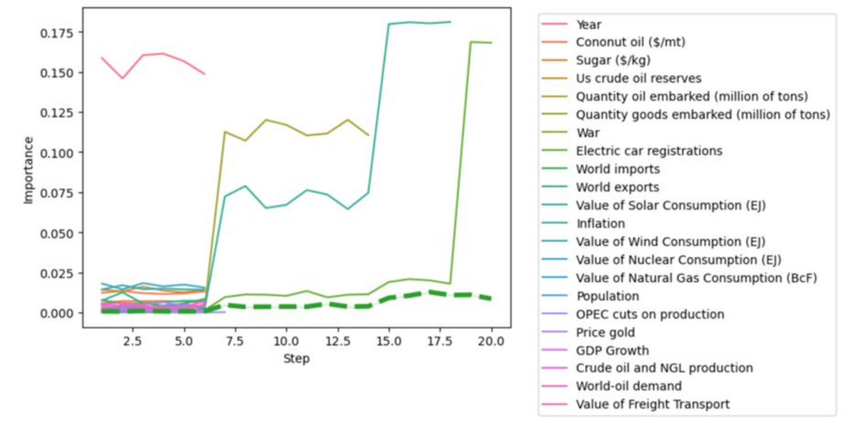
- Benchmarking process
  - evaluate effect on feature selection, ranking, algorithm performance
- P-value
- Percentile



pValue

- Lower -> stricter selection criterion -> more narrow selection of features



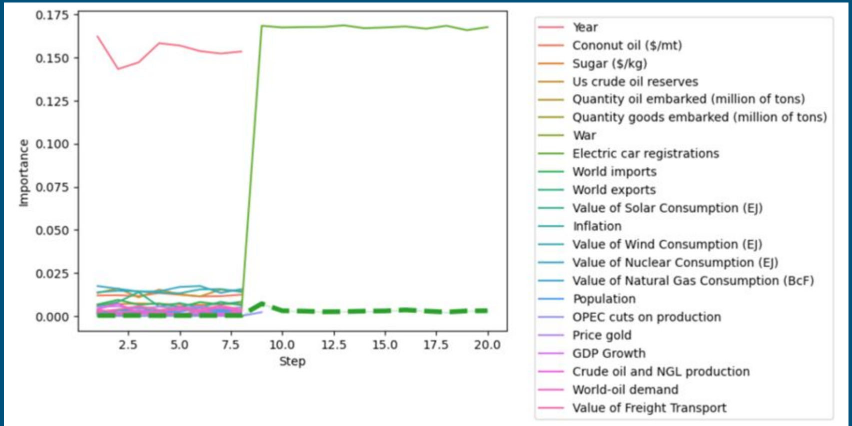


p = 0.001

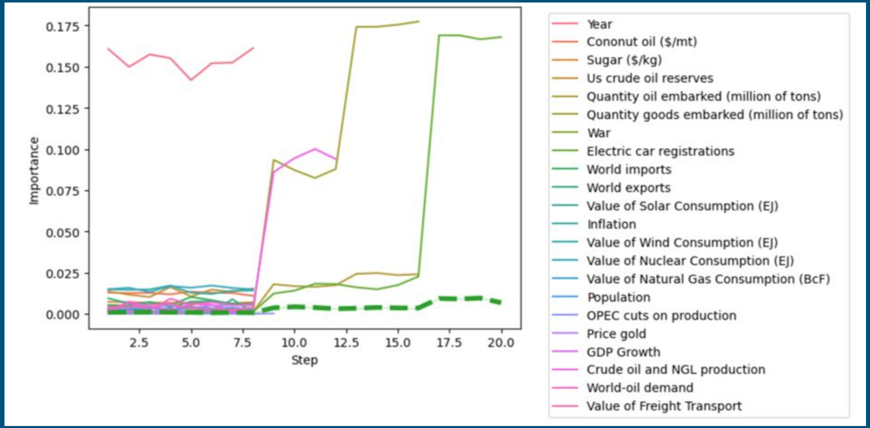
p = 0.1

Percentile

- Importance score > threshold = important feature
- Higher -> stricter selection criterion -> more narrow selection of features



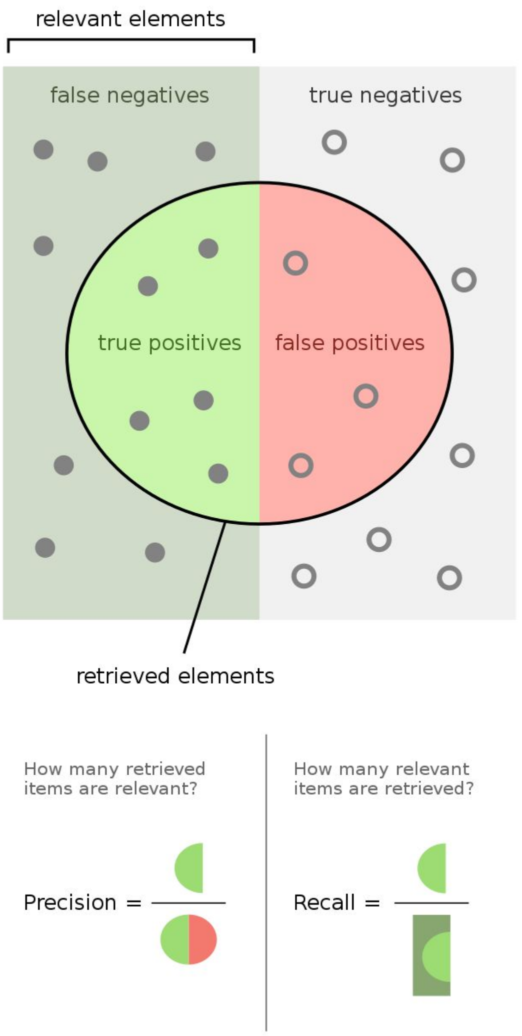
p = 70%



p = 100 %

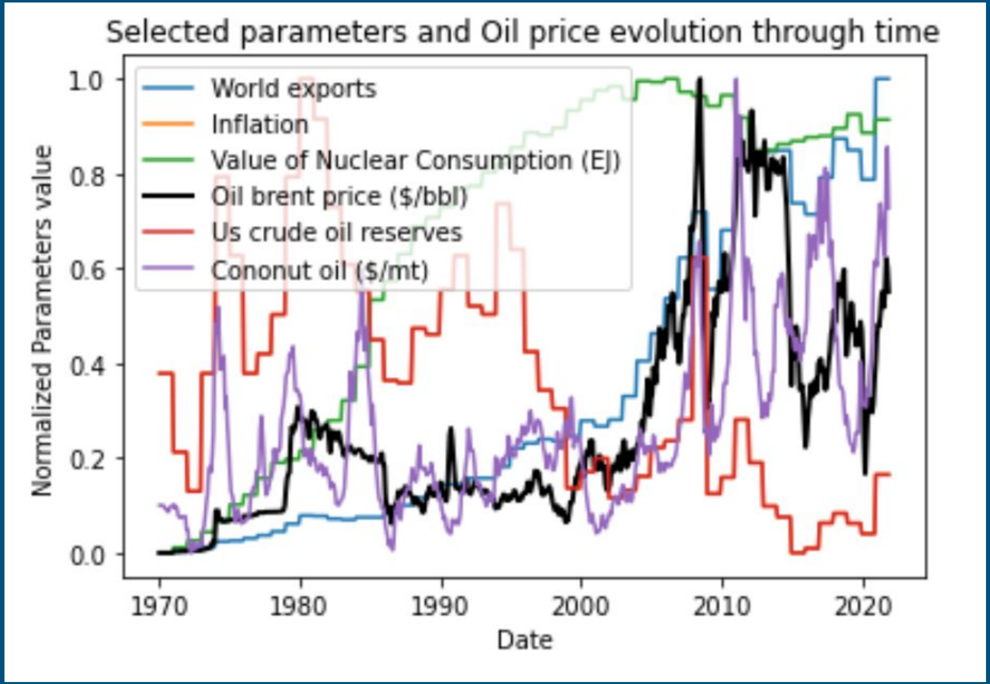
Precision and recall

- P-value
  - Influences precision, control the risk of false positives
  - Low p-value -> more precision
- Percentile
  - high percentile -> more precision, less recall (by excluding relevant features)
- Optimal values depend on data
  - Adjust parameters based on the model's performance
  - Too many false positives -> decrease the p-value
  - Lack of important features -> increase percentile
  - p-value = 0.01, percentile = 80



Selection

- Year
- Inflation
- Price of coconut oil
- US crude oil reserve
- World exports
- Value of Nuclear consumption



eBoruta: feature selection algorithm

- Selection very similar

- Reduce size
- Improve accuracy
- Improve interpretability

- SHAP useful

- Except binary values

	XGB Regressor	Random Forest
Raw Data	**23**	29
Boruta selection	21	**20**
eBoruta selection	**14**	17

Other importance measures

- SHAP required to train 2^F models
- XGBoost library
  Importance_type: 'gain', 'weight',...
- 'Total_gain' similar ranking and selection
  BUT 3s to run instead of 3:34min!!

Feature	Importance
Year	0.814146
Inflation	0.069070
Value of Freight Transport	0.044745
Cononut oil ($/mt)	0.026517
World imports	0.023654
Sugar ($/kg)	0.007412
Quantity oil embarked (million of tons)	0.003255
Quantity goods embarked (million of tons)	0.003194
Price gold	0.003064
GDP Growth	0.001655
Crude oil and NGL production	0.001197
Us crude oil reserves	0.001099
War	0.000482
Value of Nuclear Consumption (EJ)	0.000481
OPEC cuts on production	0.000024
World-oil demand	0.000005
Value of Solar Consumption (EJ)	0.000001
Population	0.000000
Value of Natural Gas Consumption (BcF)	0.000000
World exports	0.000000
Electric car registrations	0.000000
Pandemic (covid)	0.000000
Value of Wind Consumption (EJ)	0.000000

Accepted:

['Year' 'Cononut oil ($/mt)' 'Sugar ($/kg)' 'Quantity oil embarked (million of tons)' 'Quantity goods embarked (million of tons)' 'World imports' 'Inflation' 'Price gold' 'GDP Growth' 'Value of Freight Transport']

Rejected:

['Pandemic (covid)' 'War' 'Electric car registrations' 'World exports' 'Value of Solar Consumption (EJ)' 'Value of Wind Consumption (EJ)' 'Value of Nuclear Consumption (EJ)' 'Value of Natural Gas Consumption (BcF)' 'Population' 'OPEC cuts on production' 'World-oil demand']

Tentative:

['Us crude oil reserves' 'Crude oil and NGL production']

Results with total gain

New library

- Development phase
- Problems
  - Default model for continuous/discrete variables
  - Rank function
  - Test size parameter
- Changed code in the eBoruta library

Challenges

- Runtime of eBoruta
  - reduce dimension of the dataset
  - parallelize computations
  - less complex importance measure functions
- Errors
  - Debugging
- Interpretation of the results

Background material

- Never did ML before this semester
  Background in statistics and programming
  - Online tutorials and courses
  - CSE204 Introduction to Machine Learning
- Feature selection
  - Research papers
  - Textbooks

"Feature Selection for Knowledge Discovery and Data Mining" by Liu and Motoda
"Applied Predictive Modeling" by Kuhn and Johnson

- Boruta
  - Original research paper
  - SHAP documentation

Conclusion

- Usefulness of eBoruta
  - Consistent identification of significant and non-significant features
  - SHAP importance measure
  - Parameters (guide the selection)
  - Adapts to different dataset (real data)
- Alternative importance measures
- Corrected bugs in the library

What could I study after?

- Algorithms to automatically determine optimal values for parameters
- Create intuitive visualizations and summary reports to facilitate interpretation
- Explore binary features in eBoruta
- Comparing importance and performance