P2 - Diabetes prediction models

An overview of classifier algorithms

Alex Beauchamp - Otman Ezzayat Maid - Joan Gómez Català - Matija Jakovac - Álvaro Monclús Muñoz

Delivery date : 26th of May 2025

Format of presentation

1. Description of the original data
2. Preprocessing
3. Evaluation criteria of data mining models
4. Execution of different machine learning methods
   a. Naive Bayes
   b. K-NN
   c. Decision Trees
   d. Support Vector Machines
   e. Meta-Learning Algorithms
5. Comparison and conclusions

1. Description of the original data

Otman Ezzayat Maid

Presentation of dataset

- Origin: Kaggle.com
- Balanced diabetes results
  - Sourced from a survey named “Behavioral Risk Factor Surveillance System”
- 70,692 records
- Key variables:
  - Diabetes or Pre-diabetic vs Non-diabetic
  - High Blood Pressure
  - High BMI
  - Heart Conditions
  - Age
  - etc.

Joan Gómez Català

Description of metadata P1

Kaggle URL : https://www.kaggle.com/code/alexteboul/diabetes-health-indicators-dataset-notebook

Joan Gómez Català

Description of metadata P2

Kaggle URL : https://www.kaggle.com/code/alexteboul/diabetes-health-indicators-dataset-notebook

Primary objectives

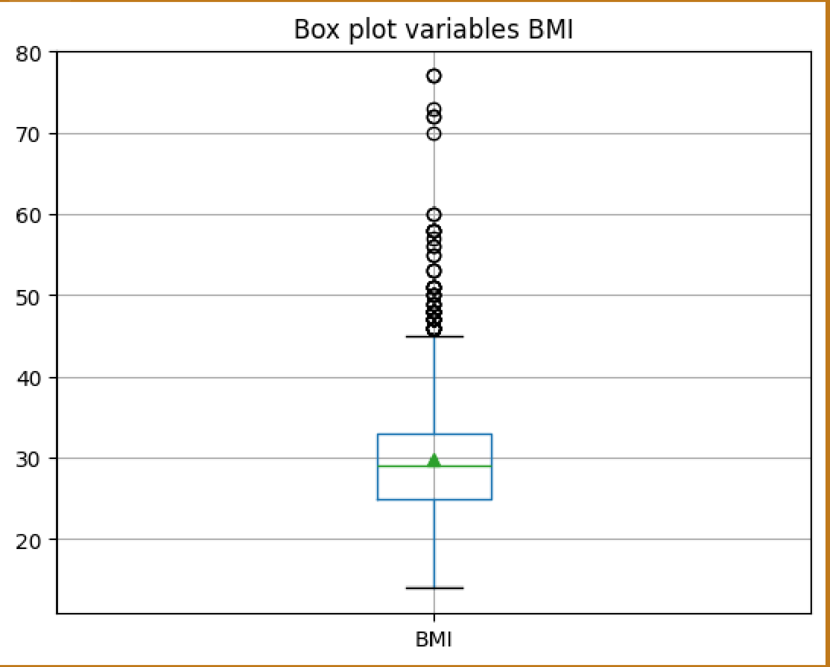
- Classify if person has diabetes
- Compute 5 different methods
    - Comparison
- Reliability of results

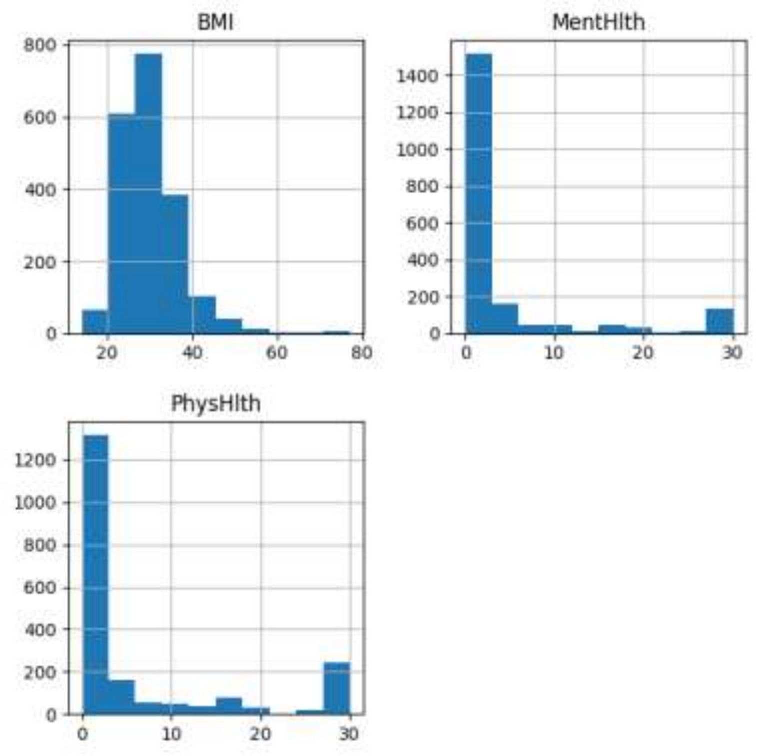
2. Preprocessing

Preprocessing steps

- Randomly select 2000 samples
    - 1000 each type
- Shortening names
    - simplicity
- Uni-variate descriptive
- Outlier detection and substitution

Uni-variate descriptive

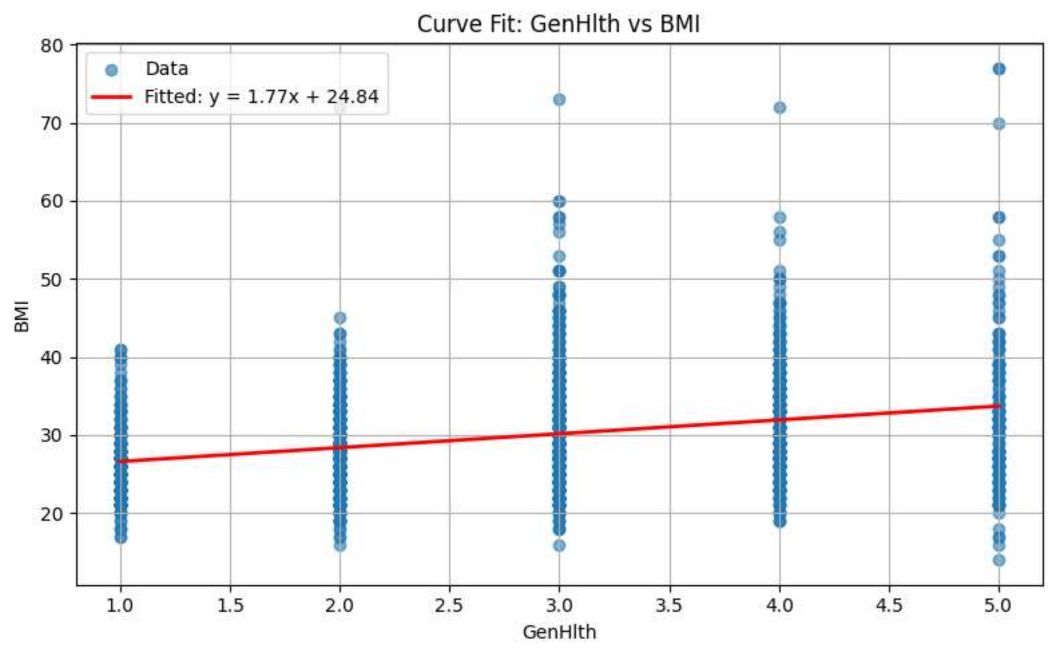




- Outliers?
  - BMI over 70

Outliers treatment

- Bi-variate analysis
- Linear regression
  - two correlated variables
- Outlier detection
  - normal distribution
- Outlier substitution
  - predicted value



X

Remaining steps

- **No missing values**
- **No text - categorical variables**
  - one-hot encoding not needed
- **No unnecessary variables**
- **Normalization not needed**
  - mostly boolean and qualitative variables

3. Evaluation criteria of data mining models

Splitting procedure

- Only once for all, right at the start
- Random_state=42 for repeatability
- Stratified sets of data (*stratify=y*)
- 70/30 split

Evaluation methods

- Mainly accuracy and f1-score
  - General indication of quality
- Focus on recall could have been good
  - Health issue
  - Not quite as dangerous as others

	precision	recall	f1-score	support
0.0	0.70	0.75	0.72	1000
1.0	0.73	0.68	0.70	1000
accuracy			0.71	2000

This example comes from Naive Bayes

Cross validation sets

- Each algorithm decides
  - Still using same test / train split
- Generally 10, varies to 5 and 20
- Depends on needs of algorithm

4. Execution of different machine learning methods

a. Naive Bayes - Methodology & Results

Decision Threshold Tuning

- Only hyperparameter used: Decision Threshold (default: 0.5).
- If probability ≥ threshold → classified as positive (diabetes).
- 20-Fold Cross val to identify best threshold, then averaged

Performance Comparison

Default:

- 72% accuracy
- 72% average f1-score

Threshold found:

- 73% accuracy
- 72.5% average f1-score

a. Naive Bayes - Limitations & Discussion

Independence Assumption

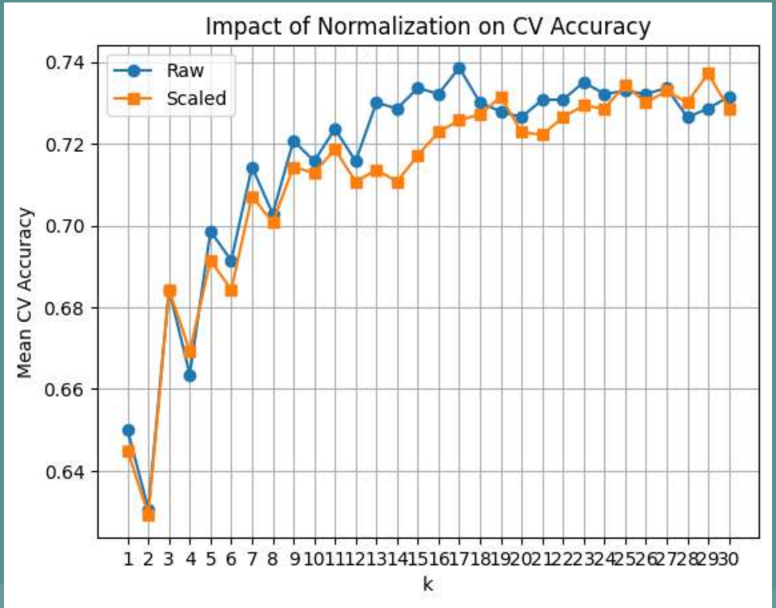
- Naive Bayes assumes feature independence
- Health indicators, like BMI or blood pressure, are often correlated
- Weakens the results as the hypothesis is not respected

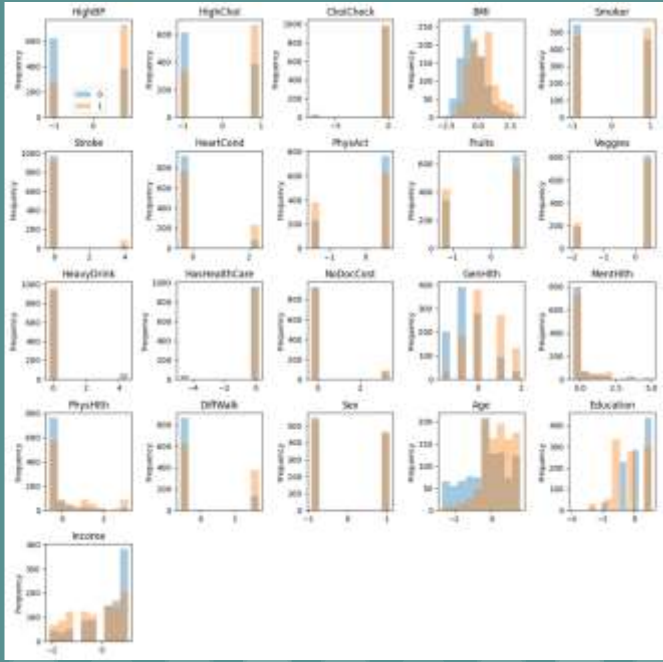
Dataset Size Consideration

- Dataset: 1000 samples per class (balanced).
- Continuous features:
  - Estimating Gaussian distributions for each feature/class combo
  - 1000 sample may be too few

b. K-NN

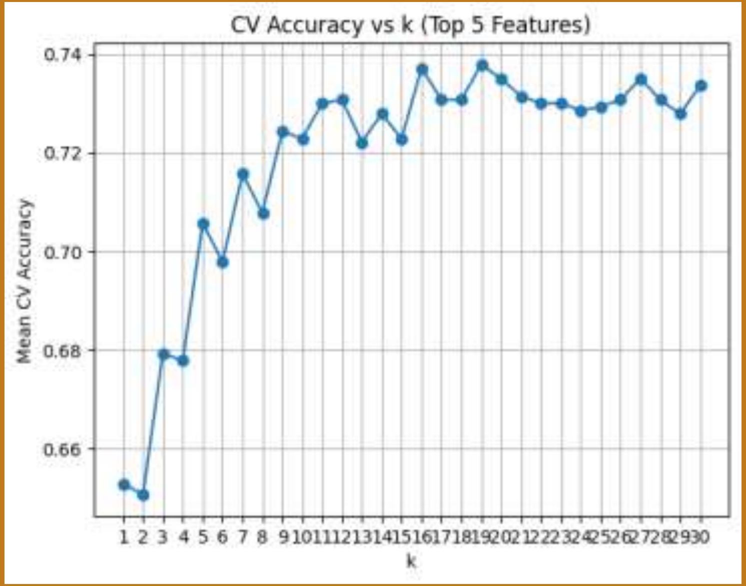
Álvaro Monclús Muñoz





b. K-NN

Álvaro Monclús Muñoz

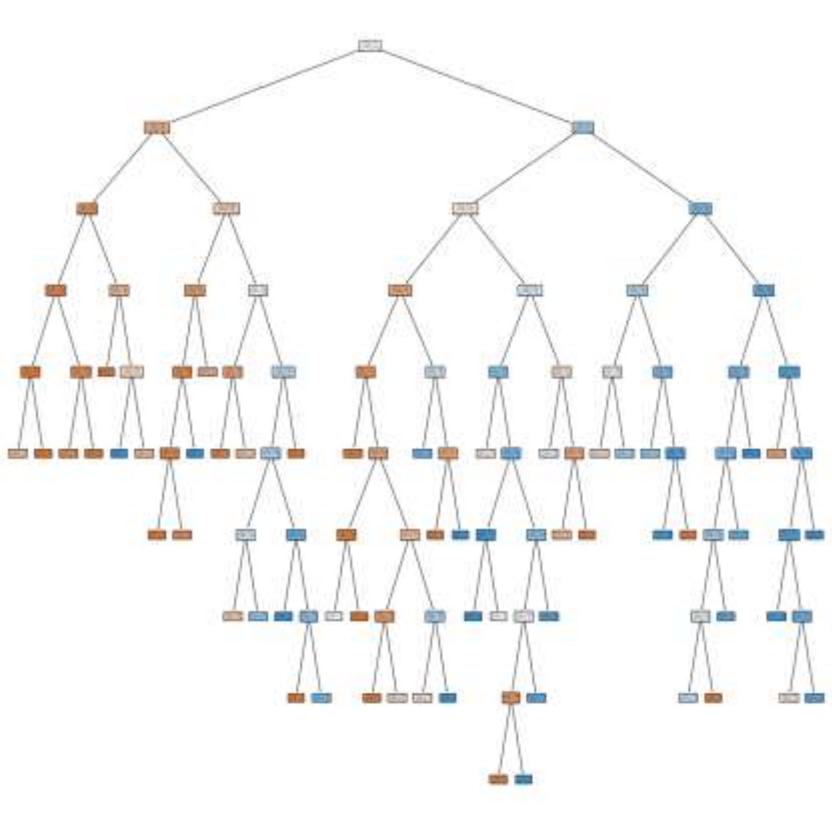



```
Metric=euclidean, weight=uniform, CV acc=0.7371
Metric=euclidean, weight=distance, CV acc=0.7407
Metric=manhattan, weight=uniform, CV acc=0.7314
Metric=manhattan, weight=distance, CV acc=0.7293
Metric=chebyshev, weight=uniform, CV acc=0.6686
Metric=chebyshev, weight=distance, CV acc=0.6736
```

c. Decision Trees

- *DecisionTreeClasifier*
  - hyperparameters
    - criterion: “entropy”
    - min_samples_split
    - min_impurity_decrease

- <u>Initial results</u>
  - big depth >10
  - large number of nodes >50
  - not interpretable
  - low accuracy - 67%

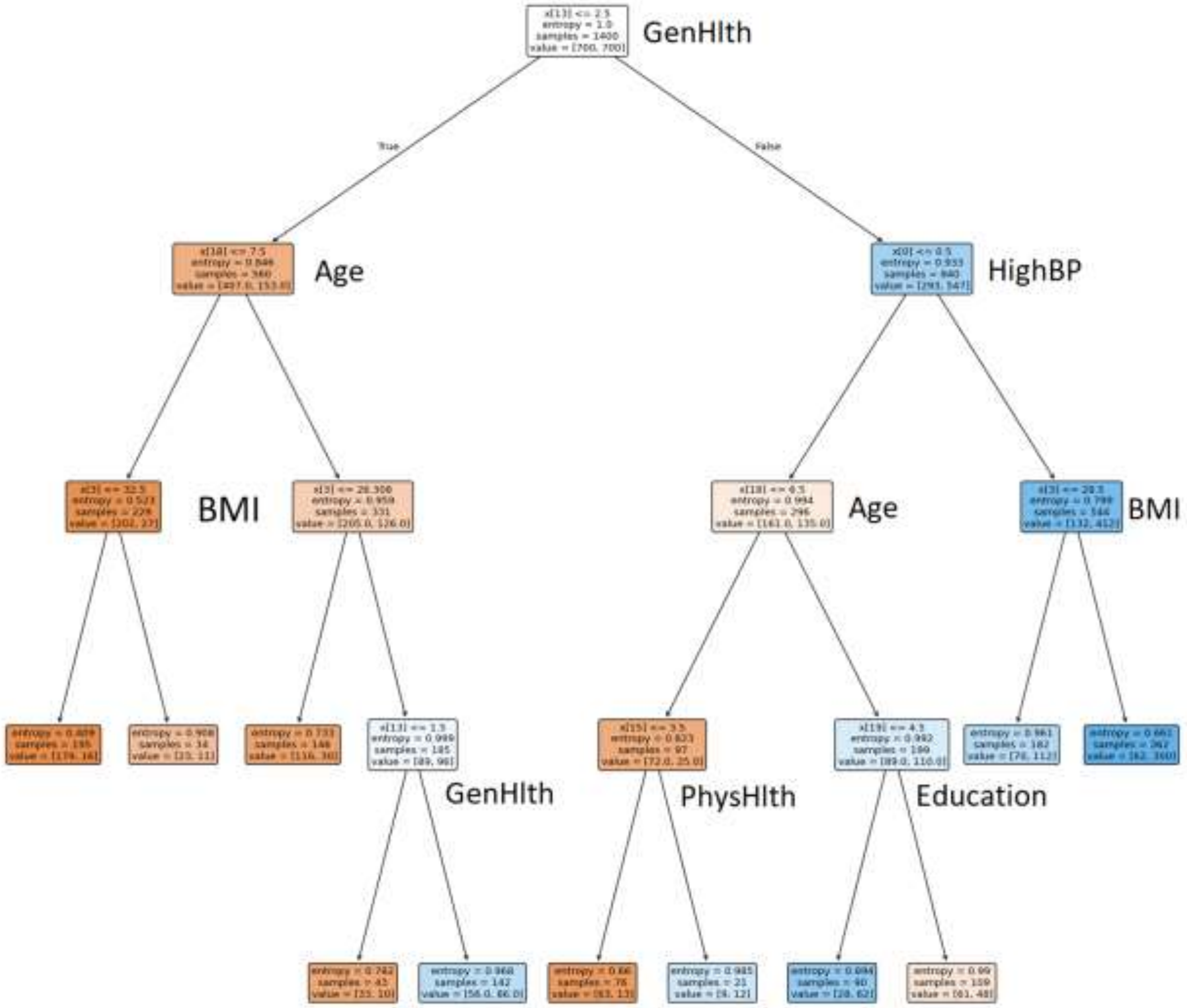


c. Decision Trees

- Grid search
  - min_samples_split
    - 2 to 20, step = 4
  - min_impurity_decrease
    - 0 to 0.05, step = 200

- Obtained hyperparameters
  - min_samples_split - 2
  - min_impurity_decrease - 0.005025

- Improved results
  - smaller depth - 4
  - smaller number of nodes - 21
  - interpretable
  - larger accuracy - 72%



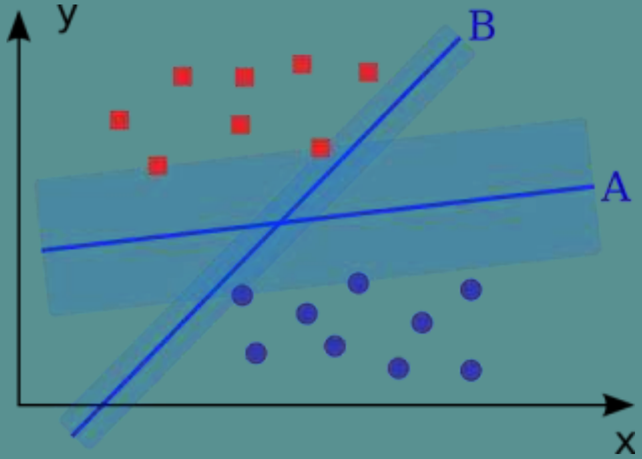
- **True Positive**
  - GenHlth ≤ 2.5 → (value: 2.0)
  - Age > 7.5 → (value: 9.0)
  - BMI > 26.31 → (value: 33.0)
  - GenHlth > 1.5 → (value: 2.0)

- BMI - common rule
- Purity of leaves
  - ideally 100%
- Errors
  - impure leaves (50%/50%)

Otman Ezzayat Maid

d. Support Vector Machines

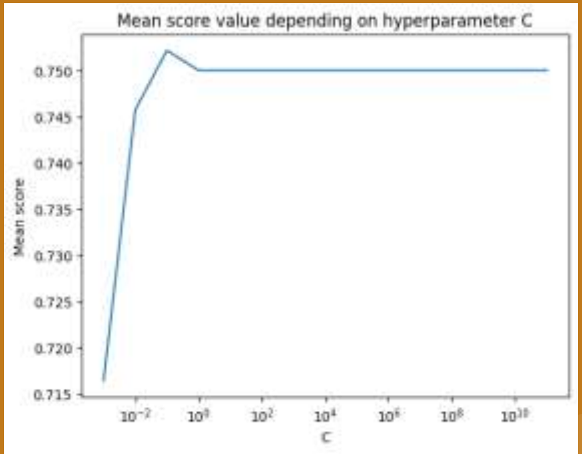
- Tested 3 kernels: Linear, Polynomial, RBF
- Balanced dataset, standardized preprocessing
- Two-step model selection for polynomial & RBF:
    - Step 1: Wide scan with 10-fold CV, log-spaced C, max_iter=100000
    - Step 2: Zoom in on best C zone, remove iteration cap
- Dataset row reduction & scaling to speed training



d. Support Vector Machines

Otman Ezzayat Maid

Linear Kernel



Parameters

- Used LinearSVC() for faster training
- Full C grid: logspace(-3, 11, 15)

Results

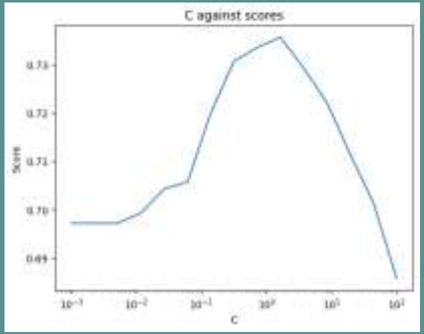
- Best C = 0.1 → 74.9% CV accuracy, 75.2% test accuracy
- Precision (class 1): ~73%, 76% average f1-score
- 846 support vectors (~60% of training set)

d. Support Vector Machines

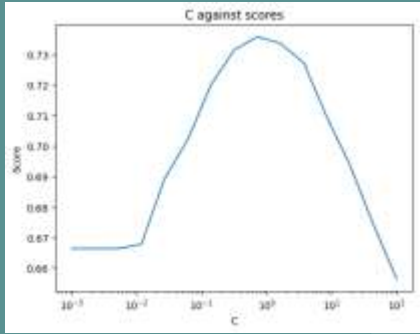
Otman Ezzayat Maid

Polynomial Kernel

Parameters



Used two-step training with 10-fold CV



Results Quadratic

Best C ≈ 1.64 → 73.6% CV, 73% test accuracy

851 supports (754 slack)

Results Cubic

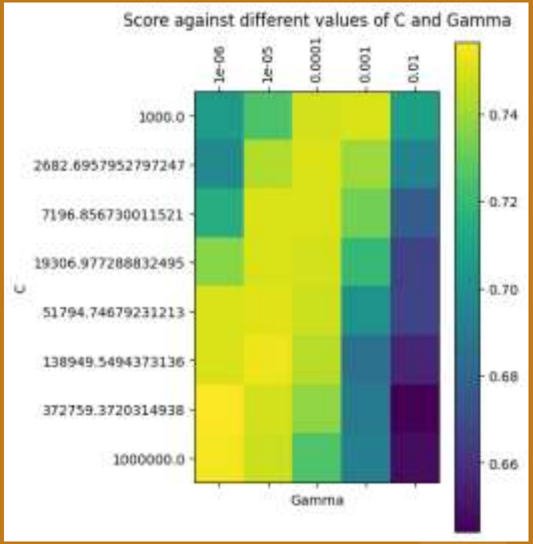
Best C ≈ 0.72 → 73.6% CV, 73.5% test accuracy

932 supports (764 slack)

Otman Ezzayat Maid

d. Support Vector Machines

RBF Kernel



Parameters

- Two-step grid search on C and gamma
- Step 1: C ∈ [0.1, 10^6], gamma ∈ [10^{-6}, 10], with iteration cap
- Step 2: narrowed region, no iteration cap

Results

- Best C ≈ 3.7×10^5, gamma = 10^{-6}
- 75.6% CV accuracy, 75.7% test accuracy
- Highest accuracy, highest complexity

Joan Gómez Català

e. Meta-learning algorithms

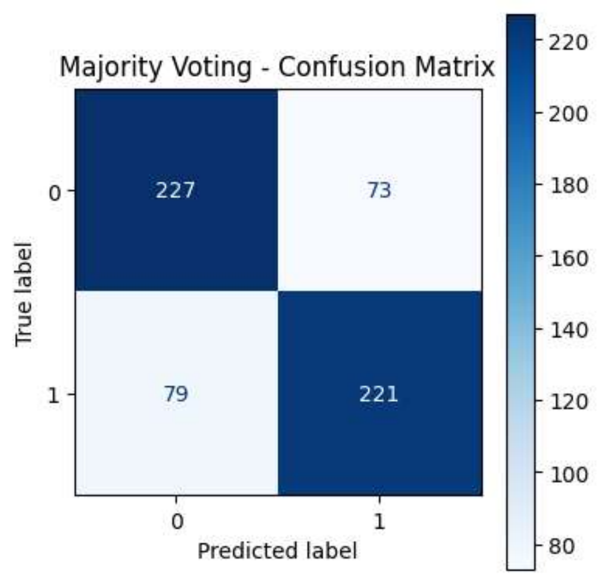
Majority Voting

Parameters

- Hard voting with Naive Bayes, tuned K-NN and a Decision Tree
- 50-fold CV
- Unweighted voting

Results

- Train CV Accuracy: 0.715 [Naive Bayes]
- Best Params fo Knn: {'n_neighbors': 21, 'weights': 'uniform'} - Accuracy: 0.743
- Train CV Accuracy: 0.731 [Knn (3)]
- Train CV Accuracy: 0.660 [Dec. Tree]
- Train CV Accuracy: 0.729 [Majority Voting]



e. Meta-learning algorithms

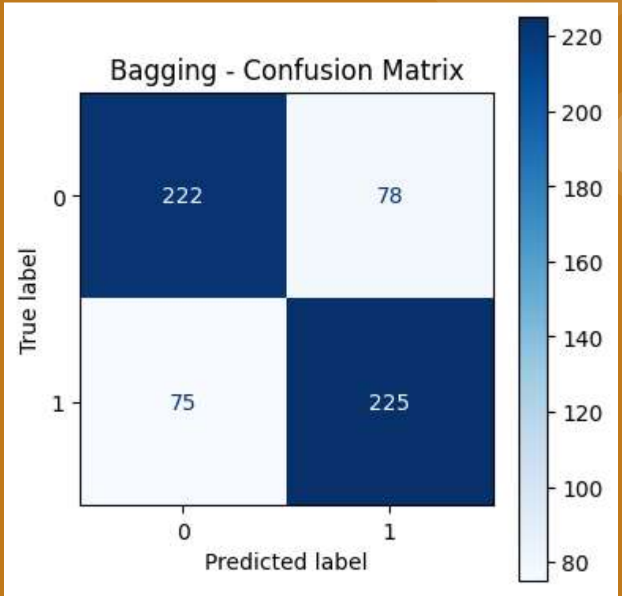
Bagging

Parameters

- Base estimator: Decision Tree
- Max features: 0.35
- 10-fold CV

Results

- Accuracy: 0.629 [n° estimators: 1]
- Accuracy: 0.721 [n° estimators: 50]
- Accuracy: 0.720 [n° estimators: 100]
- **Accuracy: 0.732 [n° estimators: 200]**



Joan Gómez Català

e. Meta-learning algorithms

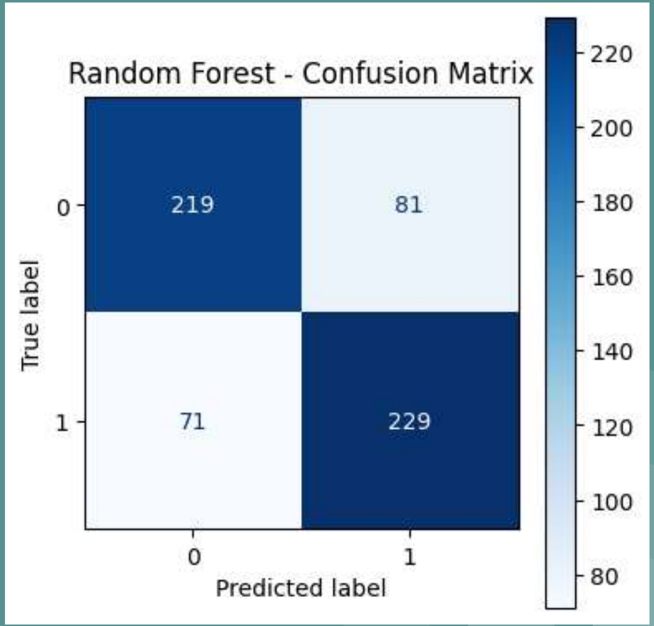
Random Forest

Parameters

- 10-fold CV

Results

- Accuracy: 0.644 [nº estimators (trees): 1]
- Accuracy: 0.726 [nº estimators (trees): 50]
- Accuracy: 0.728 [nº estimators (trees): 100]
- Accuracy: 0.741 [nº estimators (trees): 200]



e. Meta-learning algorithms

Joan Gómez Català

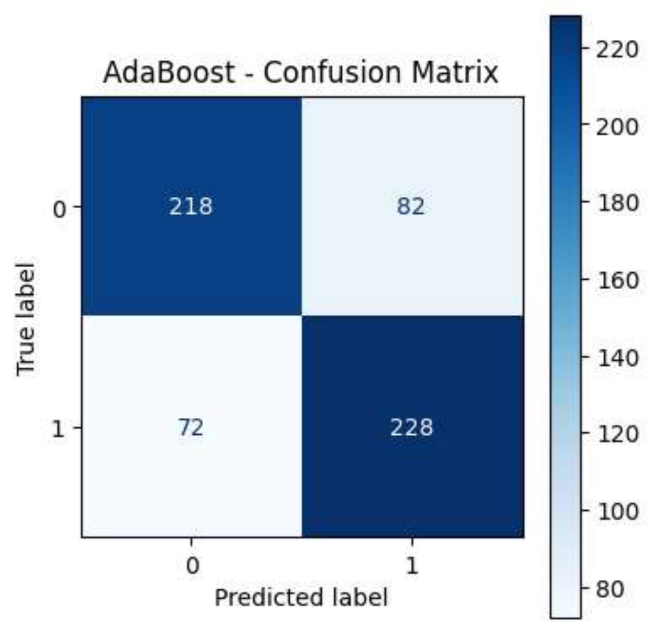
AdaBoost

Parameters

- ???

Results

- Accuracy: 0.668 [nº estimators: 1]
- Accuracy: 0.734 [nº estimators: 50]
- Accuracy: 0.733 [nº estimators: 100]
- Accuracy: 0.736 [nº estimators: 200]



5. Comparison and conclusions

Final conclusions

Álvaro Monclús Muñoz

Method	**CV Accuracy**	**95% CI**
Naive Bayes (threshold tuned)	73%	68% - 78%
K-NN (k=16, top 5 features)	75%	70.3% - 80.1%
Decision trees (depth 4, pruned)	72%	66.9% - 77.1%
Linear SVM (C = 0.1)	74.9%	70.3% - 80.1%
Polynomial kernels	73.6%	68.3% - 78.9%
RBF SVM ($C \approx 3.7\times10^{5}$, $\gamma = 1\times10^{-6}$)	75.7%	70.8% - 80.6%
Majority Voting	72.9%	67.9% - 77.9%
Bagging (200 trees)	73.2%	68.2% - 78.2%
Random forest (200 trees)	74.1%	69.1% - 79.1%
AdaBoost (200 estimators)	73.6%	68.6% - 78.6%

Questions