Regularization

Embedded Methods - Regularization

- רגולריזציה - הגבלה של המודל הגורמת לבחירת מאפיינים חזקים בלבד
- bias & variance ע"י איזון בין overfittingרגולריזציה באה לתת מענה ל-
- עד כדי כך שהוא trainמצב זה נגרם כאשר האלגוריתים מאומן חזק על ה- תופס גם את הדוגמאות שאינם מייצגות את התכונות האמיתיות של כלל ה- data("רעשים")
- לימוד של המודל על הרעשים מצד אחד עושה אותו גמיש כך שיתפוס את overfit אבל במחיר של trainכל הנקודות ב-

Embedded Methods - Regularization

- רגולריזצה ב- Linear Regression Models
- רגולריזציה ב- SVR / SVC
- רגולריזציה ב- Random Forest

LASSO ,Ridge Regression Embedded Methods

Linear regression under common regularizations

- Ridge regression: L2 regularization

$$\min_{w}: \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - H\left(w, x^{(i)}\right)\right)^2 + \frac{\lambda}{2}\|w\|_2^2$$

λ is a regularization hyper parameter

- LASSO (Least Absolute Shrinkage & Selection Operator): L1 regularization

$$\min_{w}: \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - H\left(w, x^{(i)}\right)\right)^2 + \frac{\lambda}{2}\|w\|_1$$

λ is a regularization hyper parameter

- Elastic Net: combination of both L1 & L2 regularization

$$\min_{w}: \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - H\left(w, x^{(i)}\right)\right)^2 + \frac{\lambda_2}{2}\|w\|_2^2 + \frac{\lambda_1}{2}\|w\|_1$$

LASSO ,Ridge Regression
דגשים - Embedded Methods

- יכול לאפס את המקדמים של המאפיינים, ובכך מבוצע Lassoאלגוריתים ridge regression בניגוד ל- Feature Selection
- שני המודלים מתמודדים עם מאפיינים בעלי קורלציה, כל אחד בדרך אחרת
  - המקדמים של מאפיינים בעלי קורלציה יהיו זהיבridge regression-
  - אחד מהמאפיינים הקורלטיביים יתאפס בעוד השאר יקבלו ערכי קרובים ל-Lassoב- 0
- מתאים למקרים בהם יש מספר קטן של מאפיינים חזקים Lasso משמעותית, והאחרים קרובים ל-0
- מתאים למקרים בהם יש הרבה מאפיינים חזקיבRidge
- מכיל את שני הרגולריזציות, ומשלב את הייתרונות של שני Elastic Net האלגוריתמים

sklearn.linear_model.Lasso

class `sklearn.linear_model.` **`Lasso`** (*alpha=1.0*, *fit_intercept=True*, *normalize=False*, *precompute=False*, *copy_X=True*, *max_iter=1000*, *tol=0.0001*, *warm_start=False*, *positive=False*, *random_state=None*, *selection='cyclic'*) [source]

Linear Model trained with L1 prior as regularizer (aka the Lasso)

The optimization objective for Lasso is:

```
(1 / (2 * n_samples)) * ||y - Xw||^2_2 + alpha * ||w||_1
```

Technically the Lasso model is optimizing the same objective function as the Elastic Net with `l1_ratio=1.0` (no L2 penalty).

sklearn.linear_model.Ridge

class `sklearn.linear_model.` **`Ridge`** (*alpha=1.0*, *fit_intercept=True*, *normalize=False*, *copy_X=True*, *max_iter=None*, *tol=0.001*, *solver='auto'*, *random_state=None*) [source]

Linear least squares with l2 regularization.

Minimizes the objective function:

```
||y - Xw||^2_2 + alpha * ||w||^2_2
```

sklearn.linear_model.ElasticNet

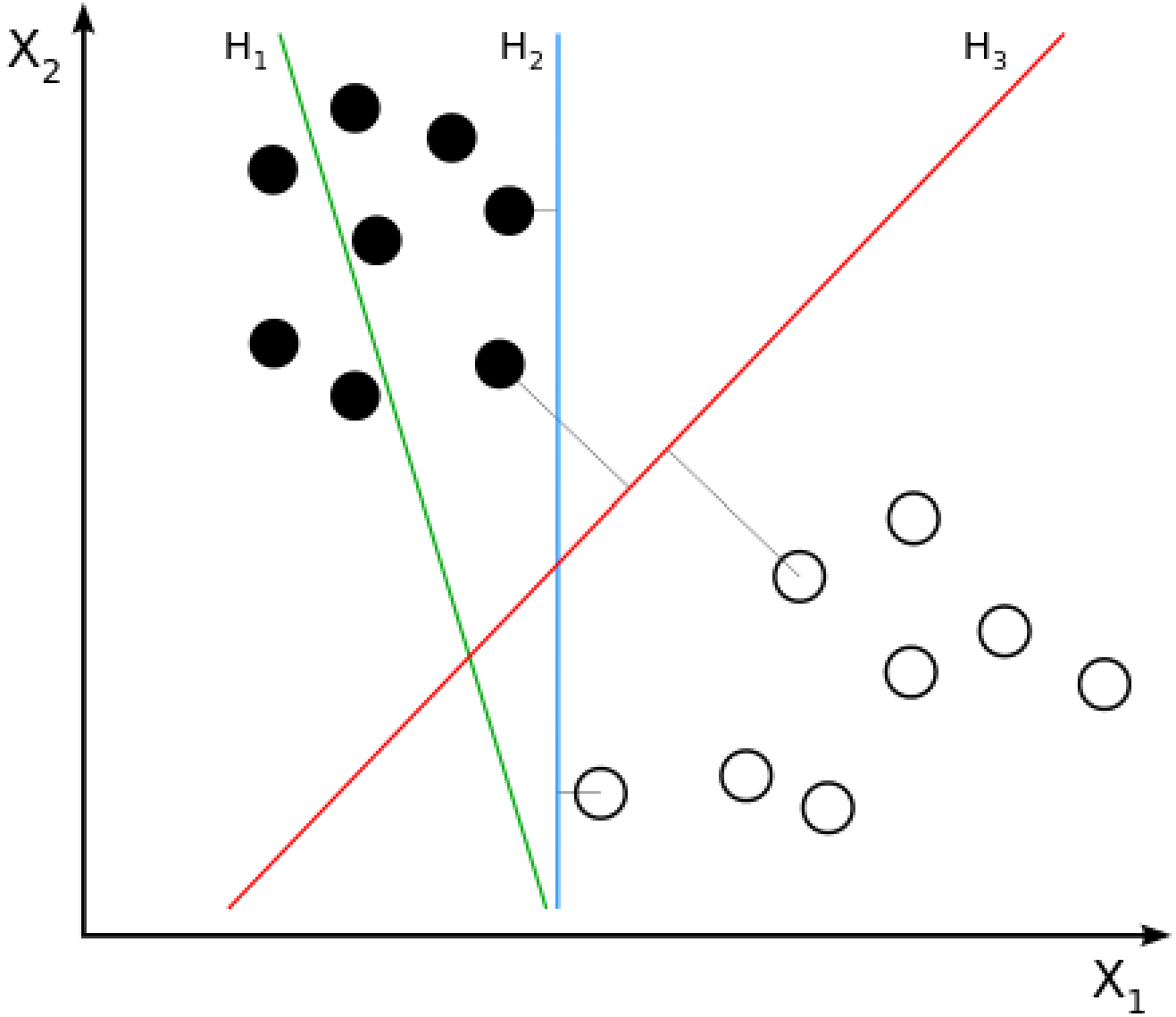
class sklearn.linear_model. **ElasticNet** (*alpha=1.0, l1_ratio=0.5, fit_intercept=True, normalize=False, precompute=False, max_iter=1000, copy_X=True, tol=0.0001, warm_start=False, positive=False, random_state=None, selection='cyclic'*) [source]

Linear regression with combined L1 and L2 priors as regularizer.

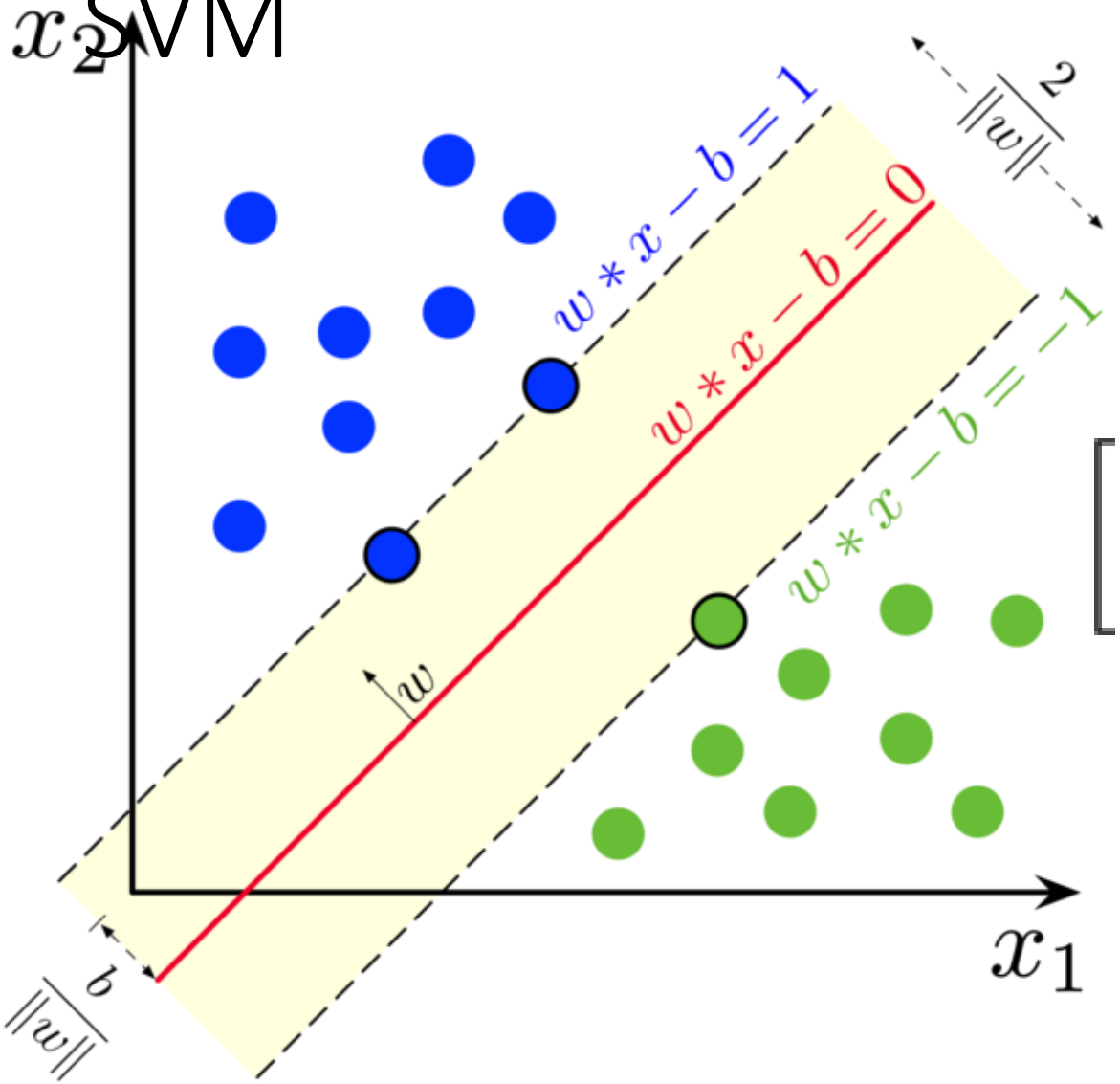
Minimizes the objective function:

```
1 / (2 * n_samples) * ||y - Xw||^2_2
+ alpha * l1_ratio * ||w||_1
+ 0.5 * alpha * (1 - l1_ratio) * ||w||^2_2
```

SVM - Support Vector Machine

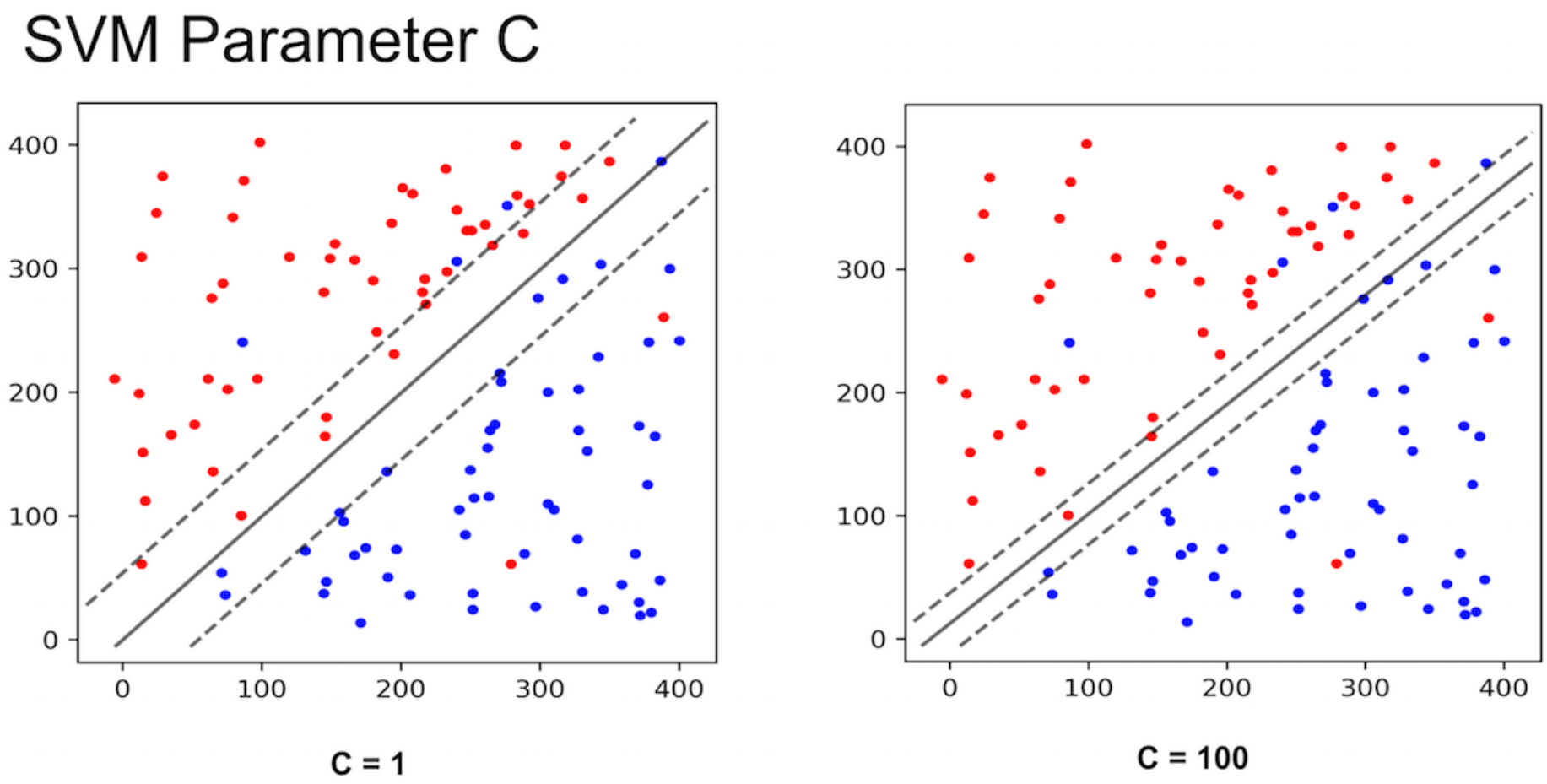


SVM



$$\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i(\vec{w}\cdot\vec{x}_i - b)\right)\right] + \lambda\|\vec{w}\|^2,$$

SVM


SVM Parameter C
400
300
200
100
0
0
100
200
300
400
C = 1
400
300
200
100
0
0
100
200
300
400
C = 100

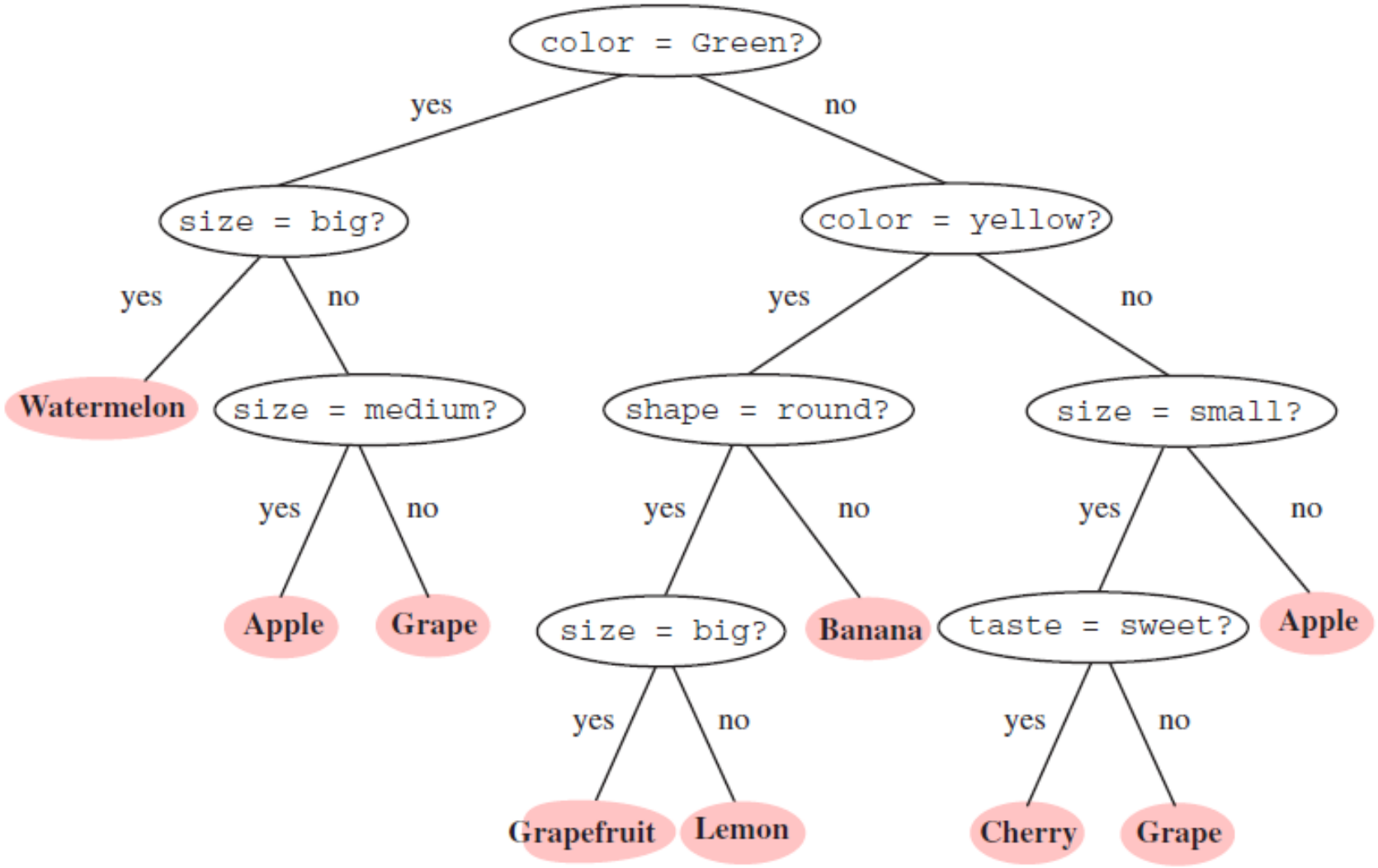
sklearn.svm.SVR

class `sklearn.svm.` **SVR** (*kernel='rbf', degree=3, gamma='auto_deprecated', coef0=0.0, tol=0.001, C=1.0, epsilon=0.1, shrinking=True, cache_size=200, verbose=False, max_iter=-1*) [source]

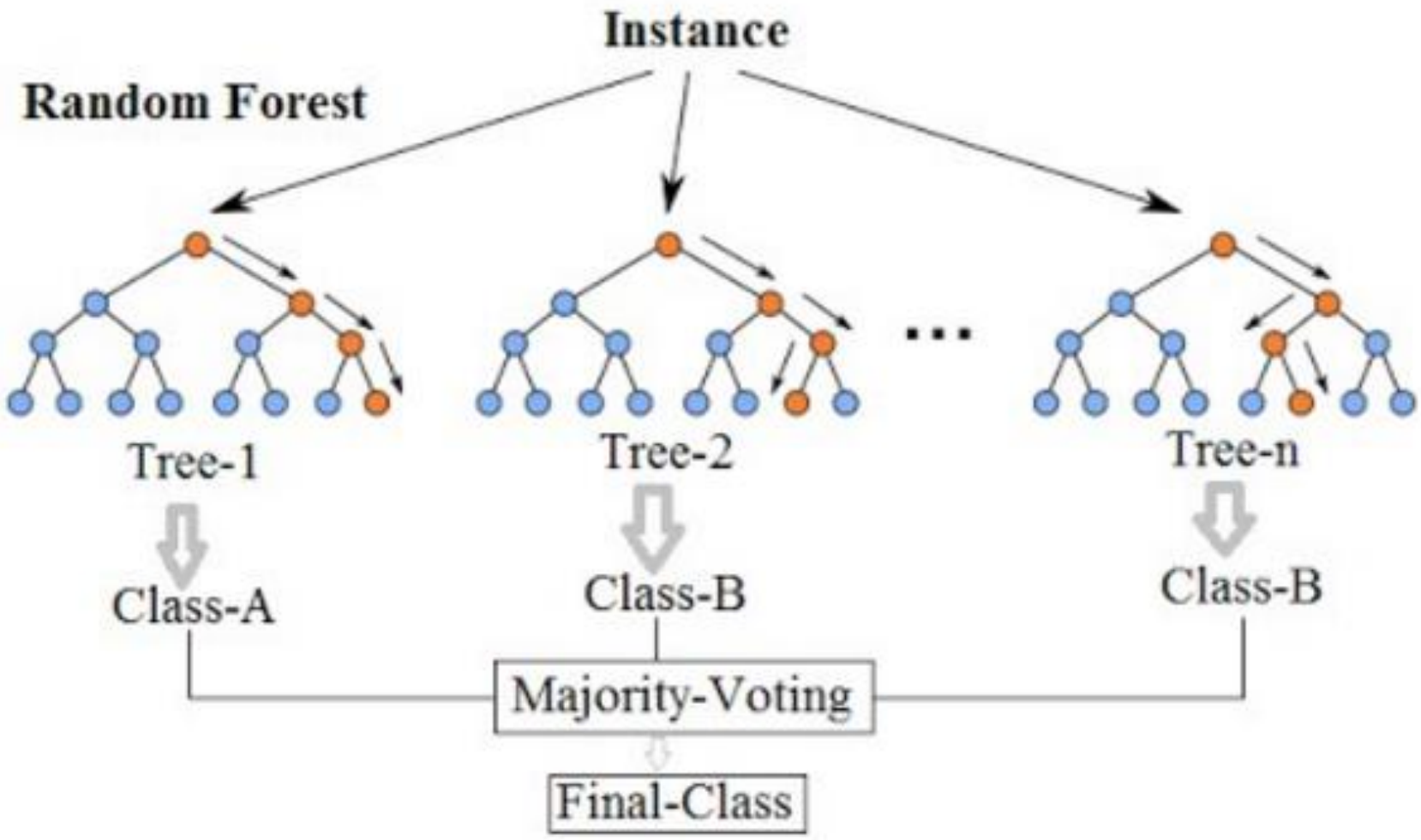
sklearn.svm.SVC

class `sklearn.svm.` **SVC** (*C=1.0, kernel='rbf', degree=3, gamma='auto_deprecated', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', random_state=None*) [source]

Decision Tree



Random Forest Simplified



על איזה פרמטרים ניתן לשלוט

- n_estimators - כמות העצים
- max_depth - עומק מקסימאלי של העצים
- min_samples_split - מספר המינימאלי של דוגמאות שיפוצלו (ניתן לרשום במספר או באחוזים)
- min_samples_leaf - מספר מינימאלי של דוגמאות שיהיו ב-עלה (ניתן לרשום במספר או באחוזים
- max_features - כמות המאפיינים שעל פיהם המודל בוחר לפצל את הצומת (המודל בוחר רנדומאלית מס' מאפיינים)

3.2.4.3.1. `sklearn.ensemble`.RandomForestClassifier

class `sklearn.ensemble.` **`RandomForestClassifier`** (*n_estimators='warn', criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None*) [source]