Text Analysis for Social Sciences in Python

Week 7: Text as Data – Supervised Methods & Model Evaluation

WINTER SEMESTER 21/22

HUYEN NGUYEN

HTTP://HUYENTTNGUYEN.COM

Last week…

PCA vs. PLS in Scikit-learn

Has anyone given a try yet at the bonus exercise?

…any questions with the materials?

Consultation Slots for Wednesday 01.12

https://docs.google.com/spreadsheets/d/1E-IFBNDNk7-5GxALoYRfK-P8GO5aYJ8v9j0uEjUsZ5M/edit?usp=sharing

Slight schedule change (starting @ 8AM)– please check again your assigned schedule!

Today's agenda

1. Supervised ML methods

- (Regression) Linear, Lasso vs. Ridge regression
- (Classification) Random Forest, Decision Tree vs. K-nearest Neighbor vs. Support Vector Machines

2. Model Evaluation

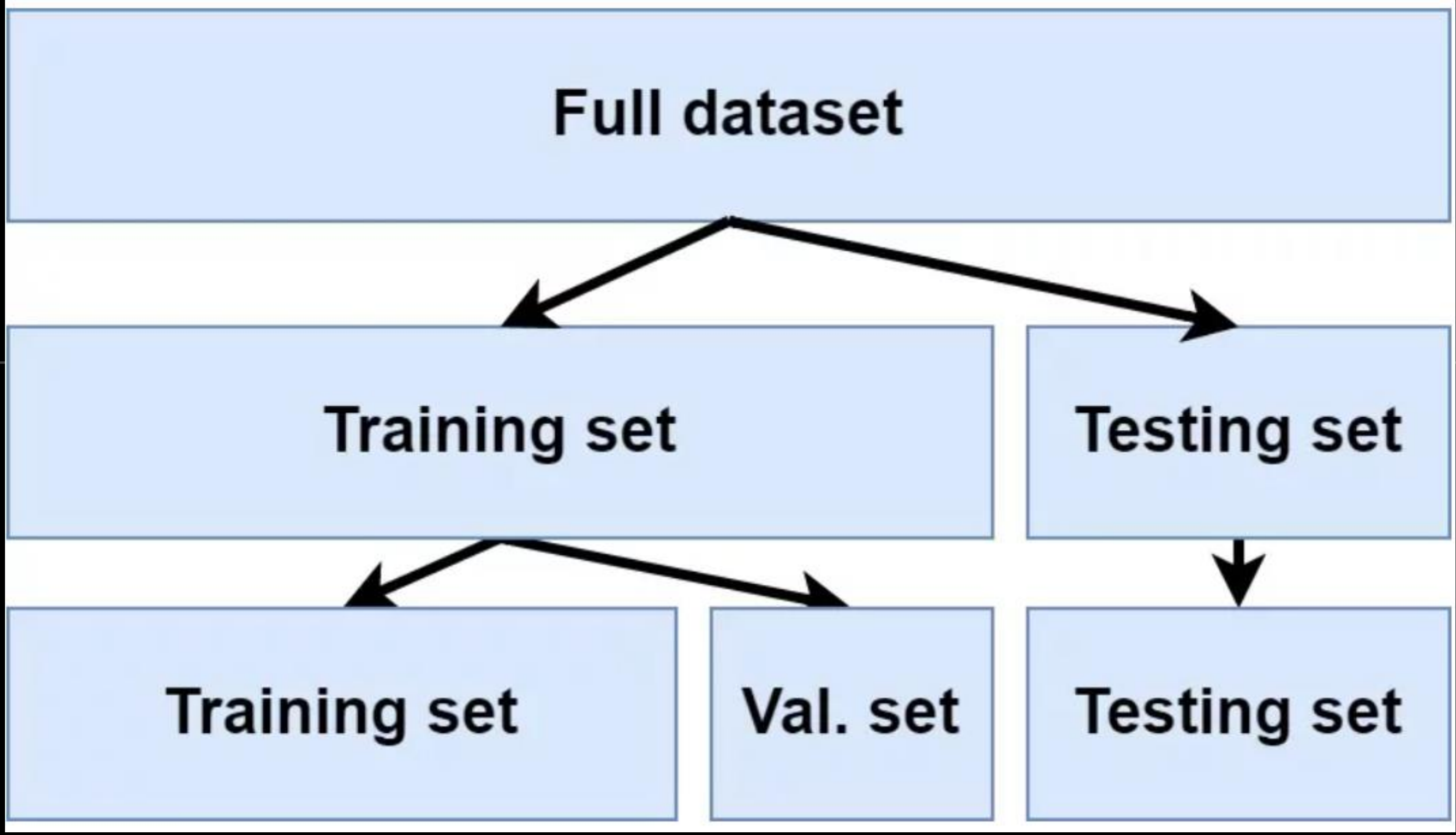
- Classification Accuracy
- Logarithmic Loss
- Confusion Matrix
- F1 score
- Mean Absolute Error (MAE) vs. Mean Square Error (MSE)

Overview – Supervised algorithms

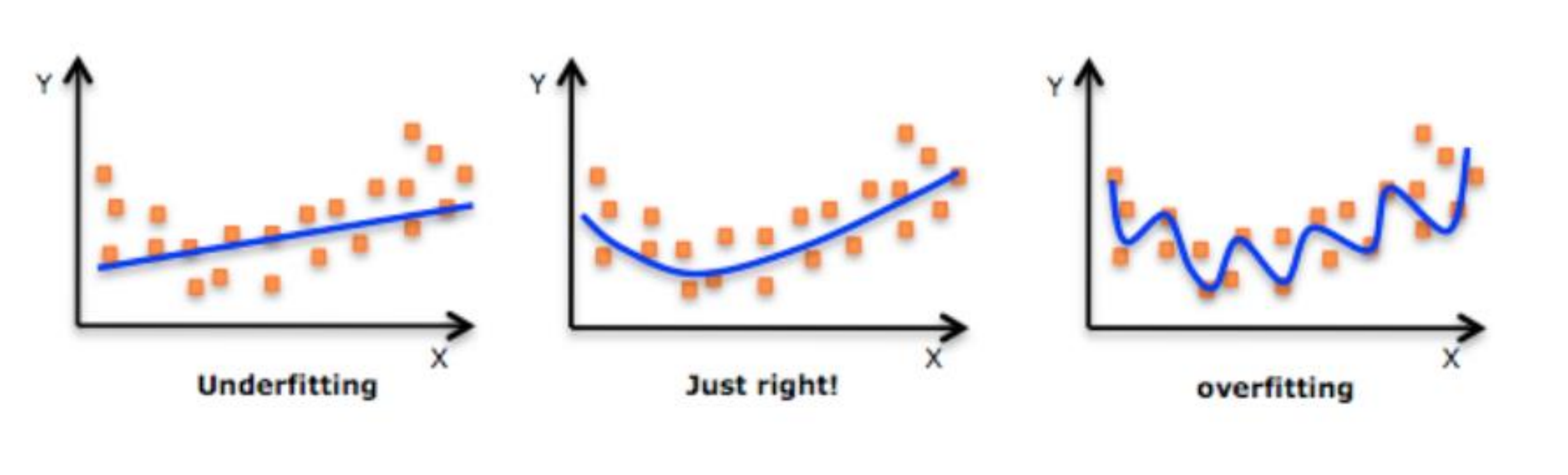
Supervised learning algorithms are used when the training data has **output** variables corresponding to the **input** variables.

The algorithm analyses the input data, learns a function to map the relationship between the input and output variables.

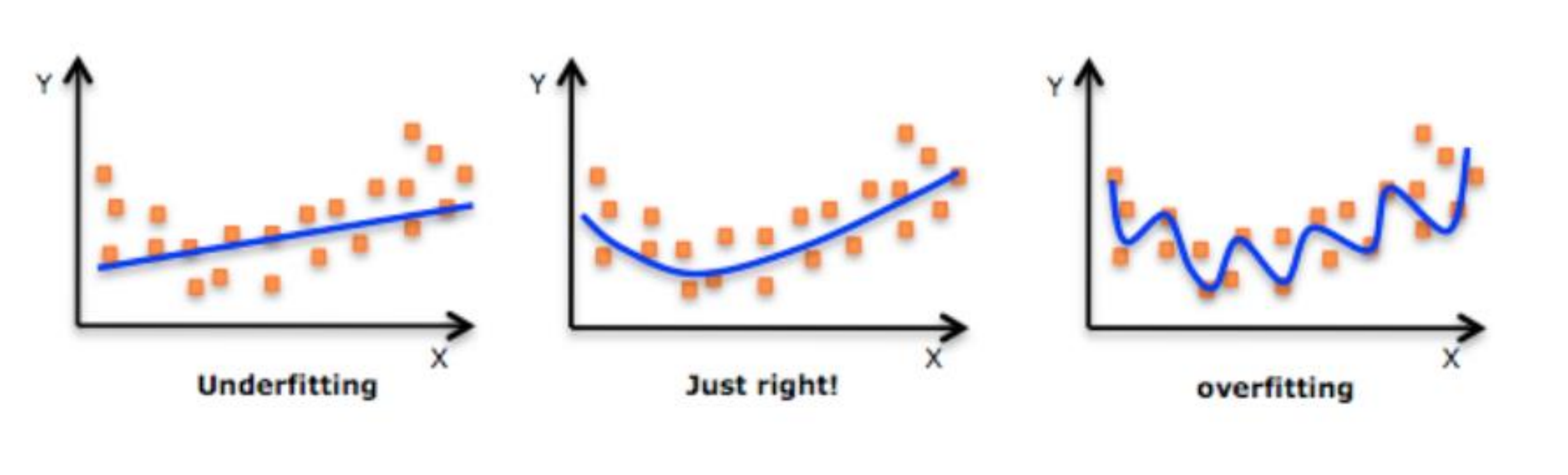
Supervised learning can further be classified into Regression, Classification, Forecasting, and Anomaly Detection (we focus on the former two in our course).



Every machine learning problem is an optimization problem.
→ <u>Goal</u>: find either a maximum or a minimum of a specific function.
→ The function that you want to optimize is usually called the <u>loss function</u> (or cost function).



How to balance the tradeoff between overfitting and underfitting a model?



How to balance the tradeoff between overfitting and underfitting a model?

→ Regularized regression

1.1 Regression Techniques

Pre-requisite: The value you want to predict is CONTINUOUS.

Key idea: These techniques adds additional information to shrink parameter values of models → induce a penalty against complexity

$$\hat{y_i} = w_0 + \sum_{j=1}^{m} \beta_j X_{ij}$$

Lasso: $J(\beta) = \sum_{i=1}^{n} (y_i - \hat{y_i})^2 + \lambda \sum_{j=1}^{m} |\beta_j|^2$

Ridge: : $J(\beta) = \sum_{i=1}^{n} (y_i - \hat{y_i})^2 + \lambda \sum_{j=1}^{m} |\beta_j|$

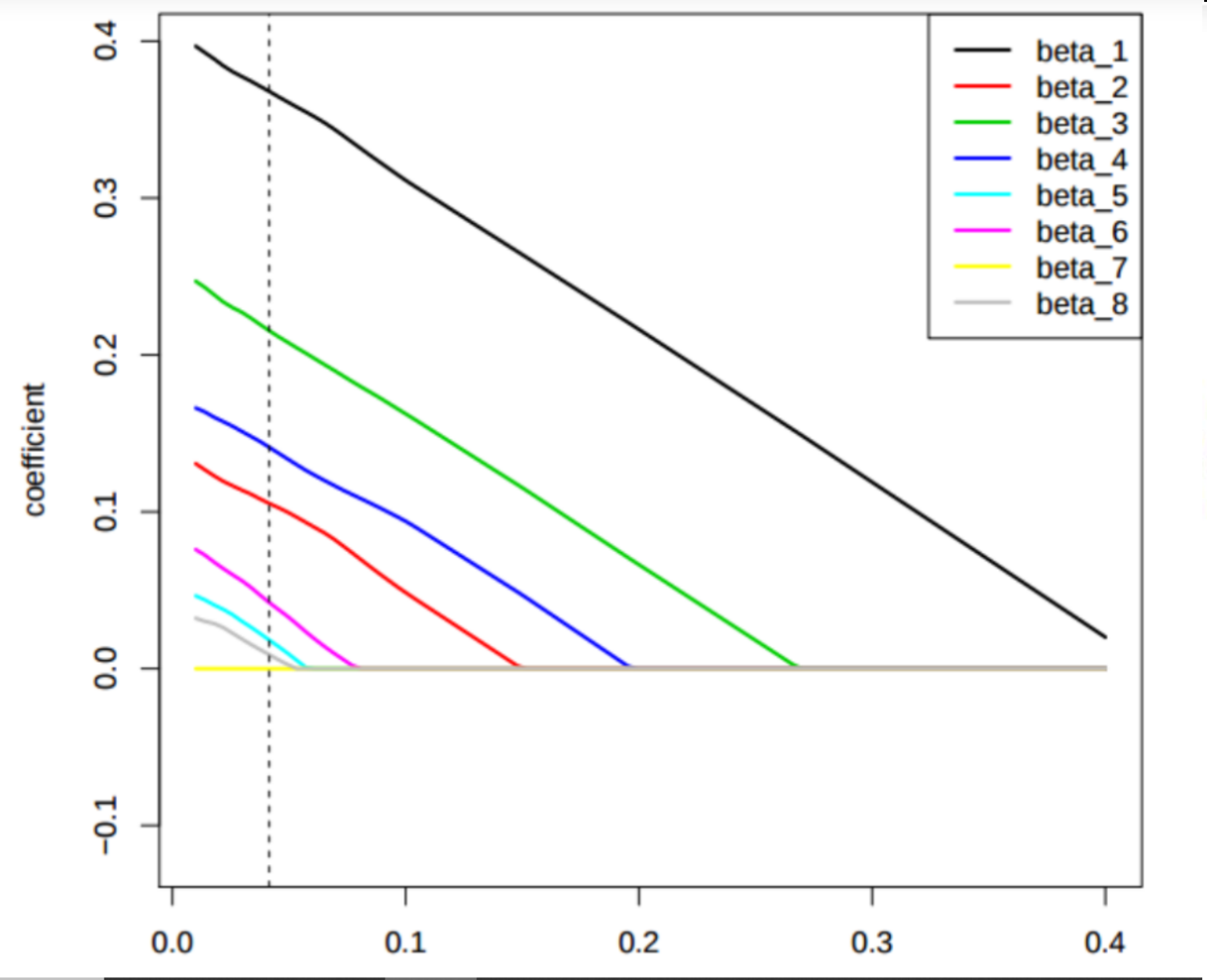
1.1 Regression Techniques (cont)

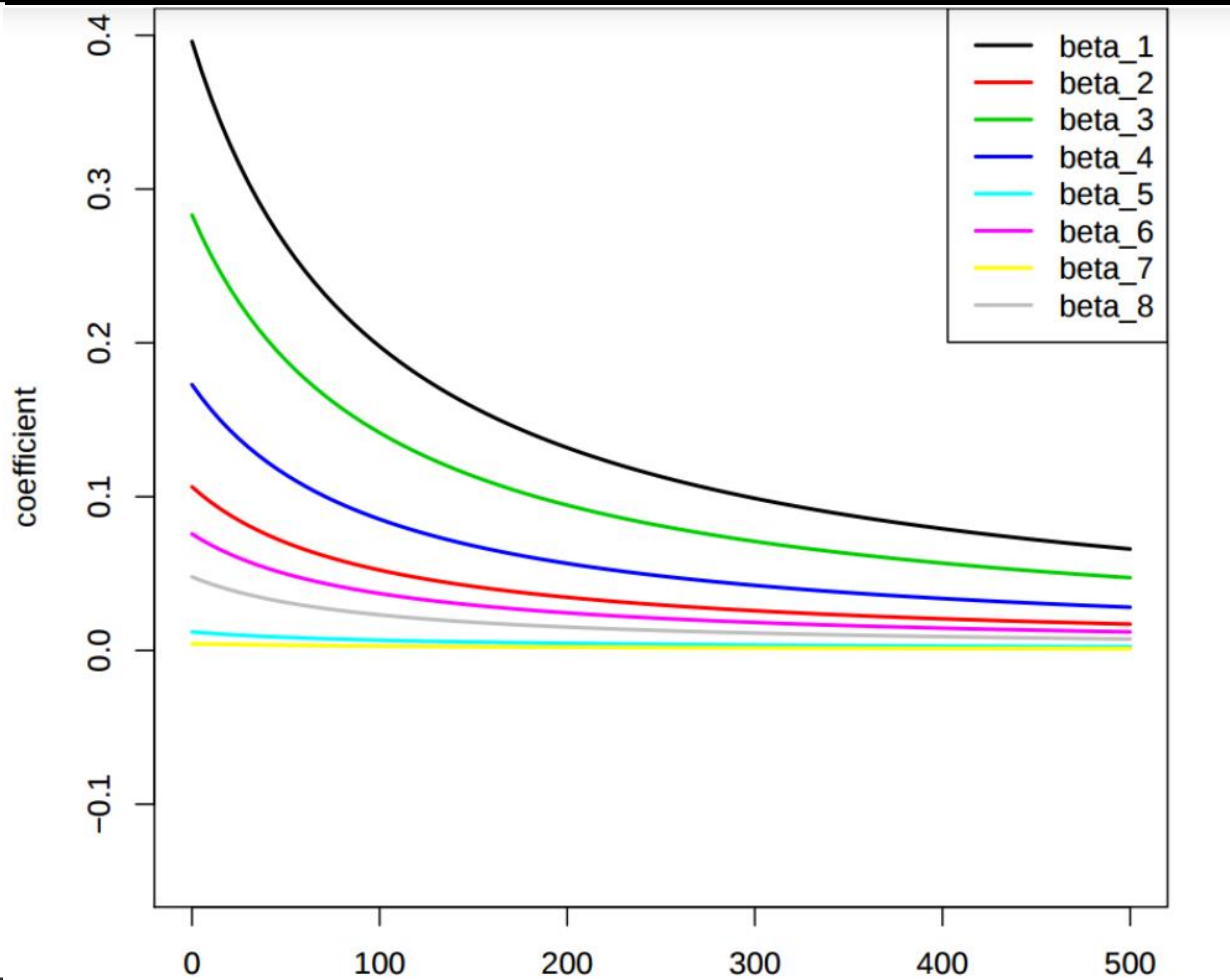
→ Both methods "punish" the loss function for high values of coefficients β, BUT…

Lasso kicks out variables with 0 coefficients (i.e. irrelevant features)

Ridge only minimize the impact of irrelevant features on the trained model, and does NOT kick them out.

Lambda plots: Lasso vs. Ridge





1.2 Classification Techniques

Pre-requisite: The value you want to predict is DISCRETE.

Key idea: Emsemble learning methods (i.e. combine predictions from multiple ML algorithms)

1.2 Classification Techniques

Pre-requisite: The value you want to predict is DISCRETE.

Key idea: Emsemble learning methods (i.e. combine predictions from multiple ML algorithms)

- Boosting = algorithms that utilize weighted averages to make weak learners into stronger learners → Each model runs, dictates what features the next model will focus on.

1.2 Classification Techniques

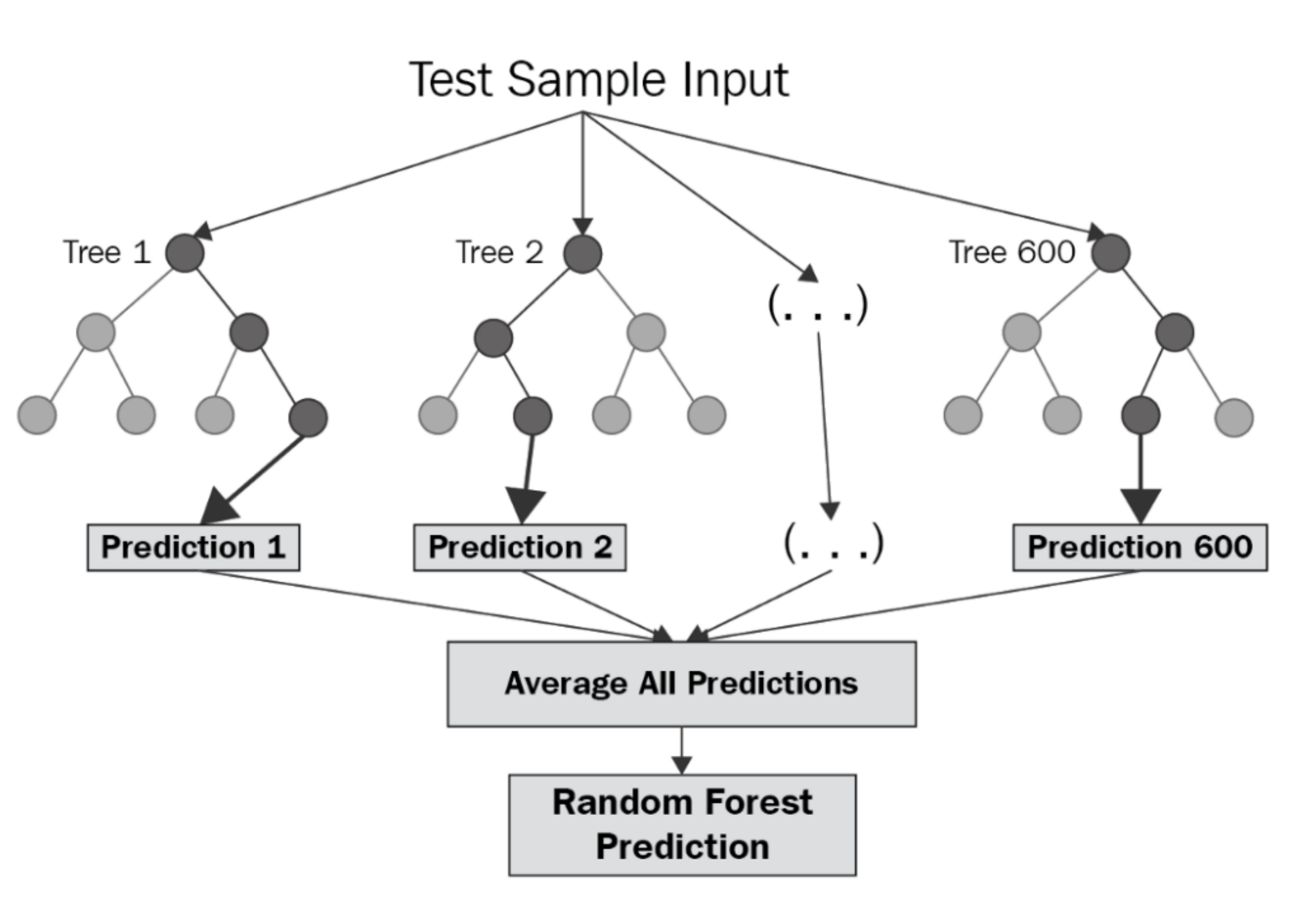
Pre-requisite: The value you want to predict is DISCRETE.

Key idea: Emsemble learning methods (i.e. combine predictions from multiple ML algorithms)

- Boosting = algorithms that utilize weighted averages to make weak learners into stronger learners → Each model runs, dictates what features the next model will focus on.
- Bagging = Each model run independently (by random sampling with replacement i.e. boostraping), and then aggregates the outputs at the end without preference to any model → Reduce the variance for algorithms with high variance.

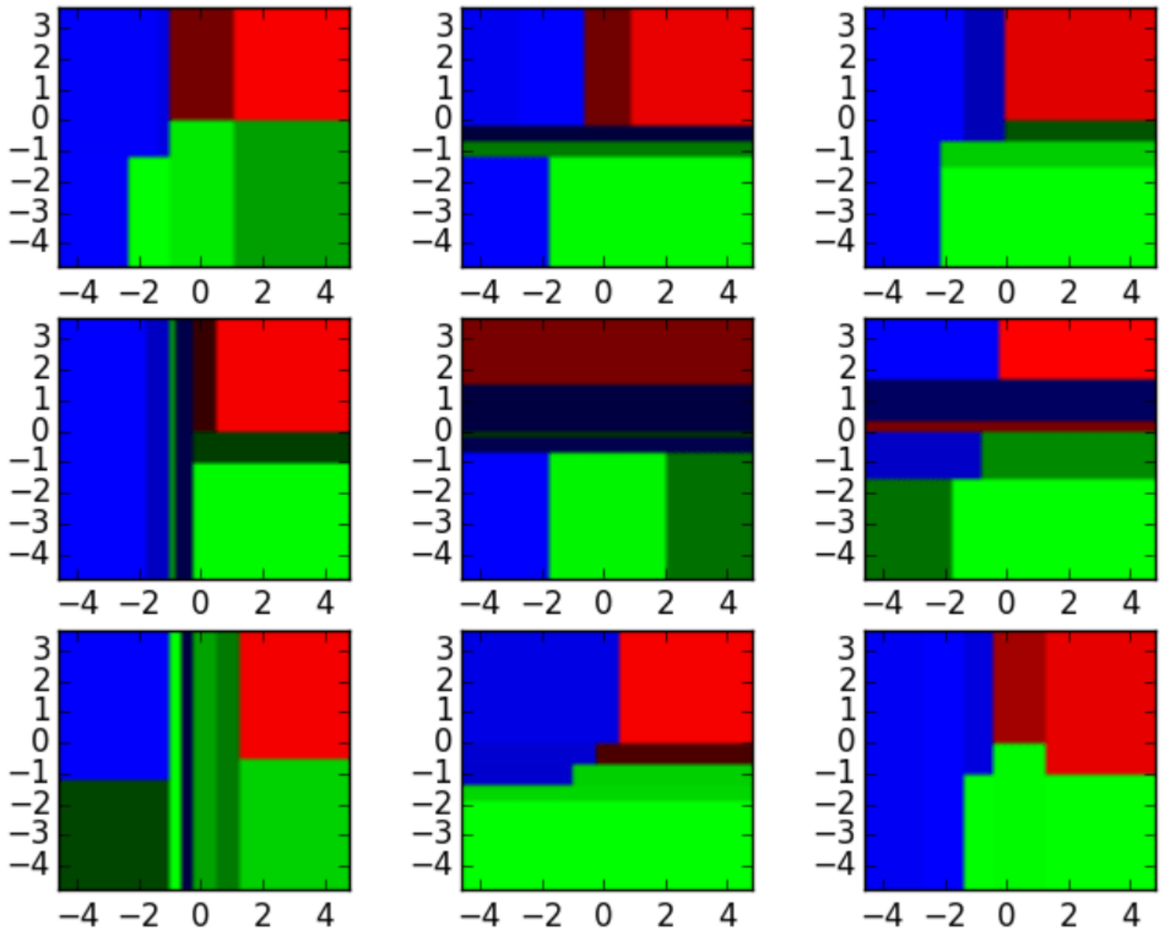
1.2 Random Forest (RF)

- Random Forest (RF) = bagging technique.
- RF constructs a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.



- RF >> Decision Tree (DT) (generally) because…
- DT = computationally expensive +
- sensitive to training data

1.2. RF = meta-estimator…



Nine Different Decision Tree Classifiers

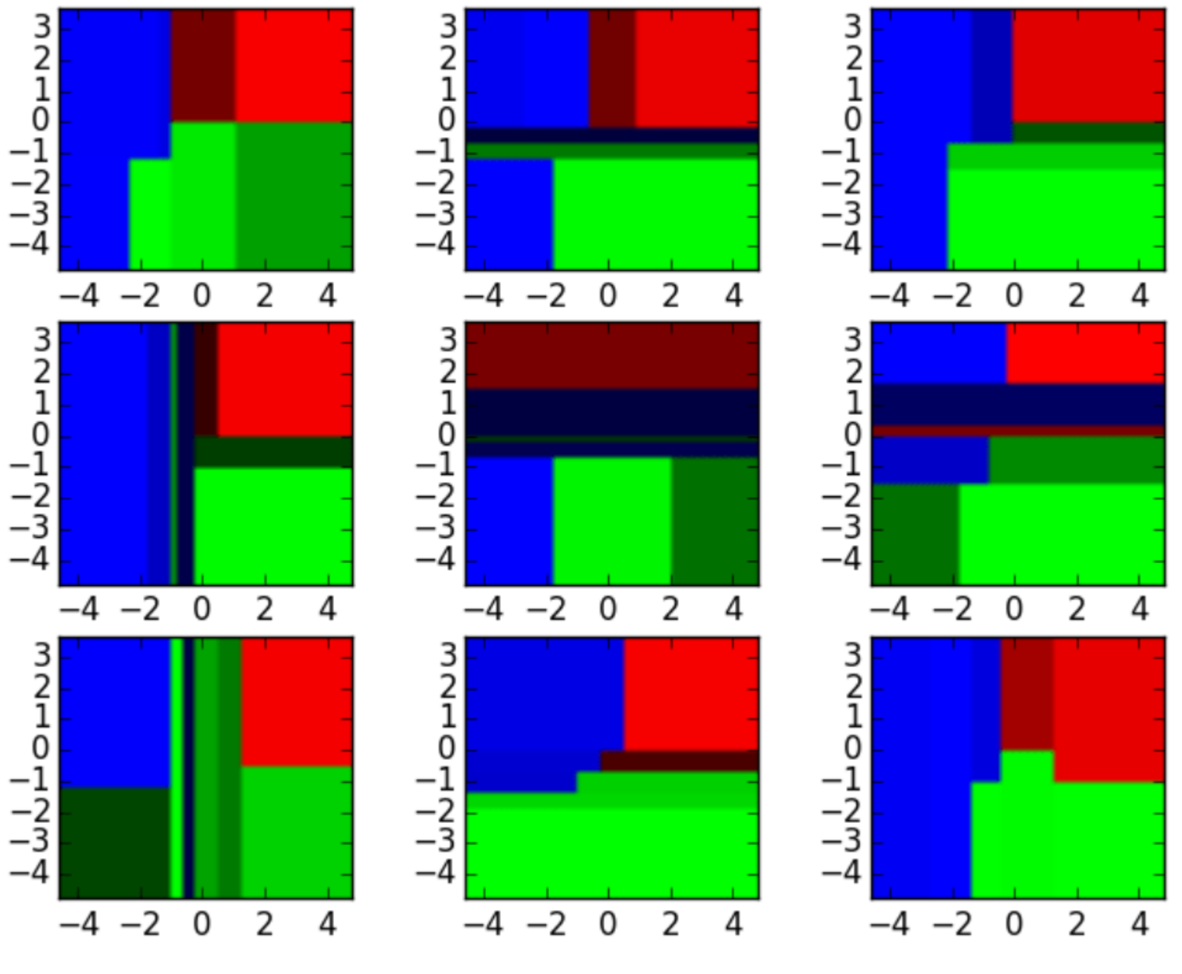
…which **aggregates many decision trees,** whereby:

- the number of features that can be split on at each node is limited to some percentage of the total (i.e. **hyperparameter**). → the ensemble model **does not rely too heavily on any individual feature + fair use of all potentially predictive features.**

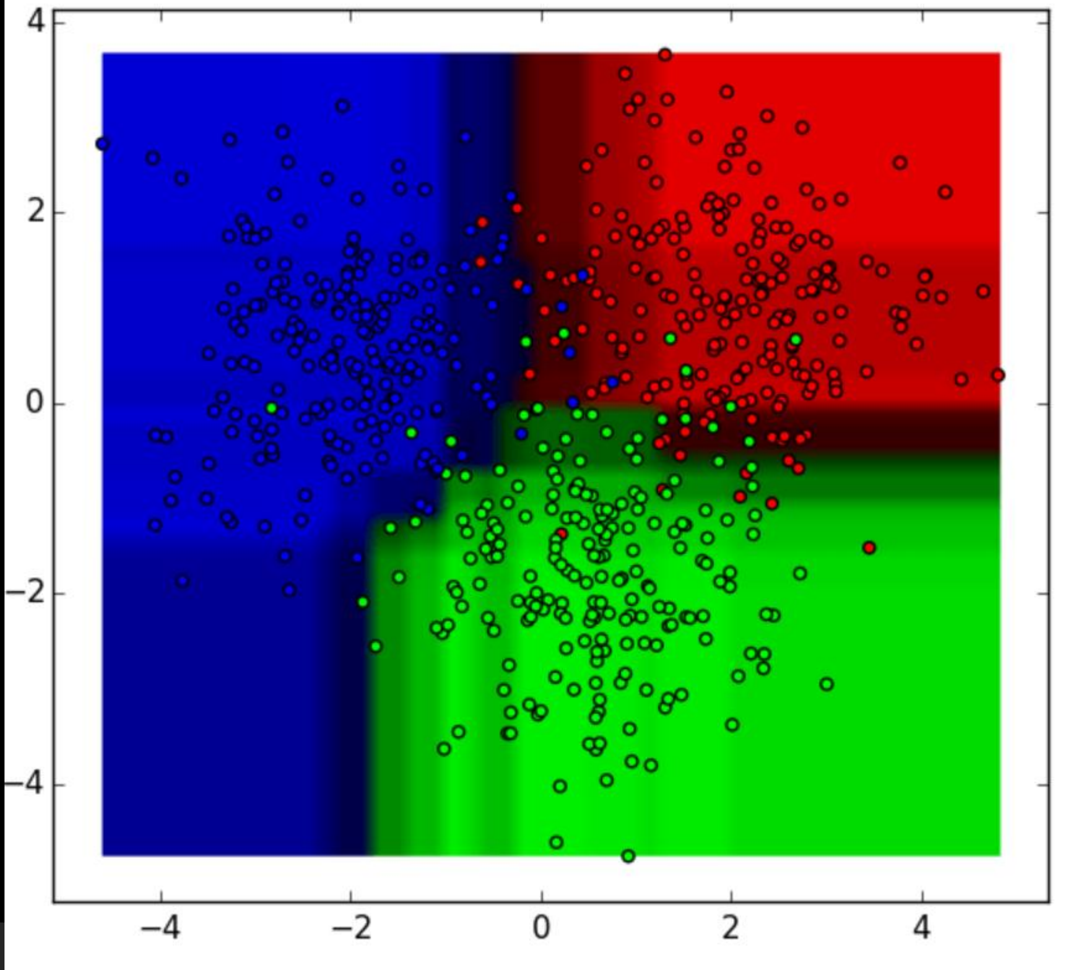
- each tree draws a random sample from the original data set when generating its splits → adding a further element of randomness → prevents **overfitting.**

Read more on RF @

https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f



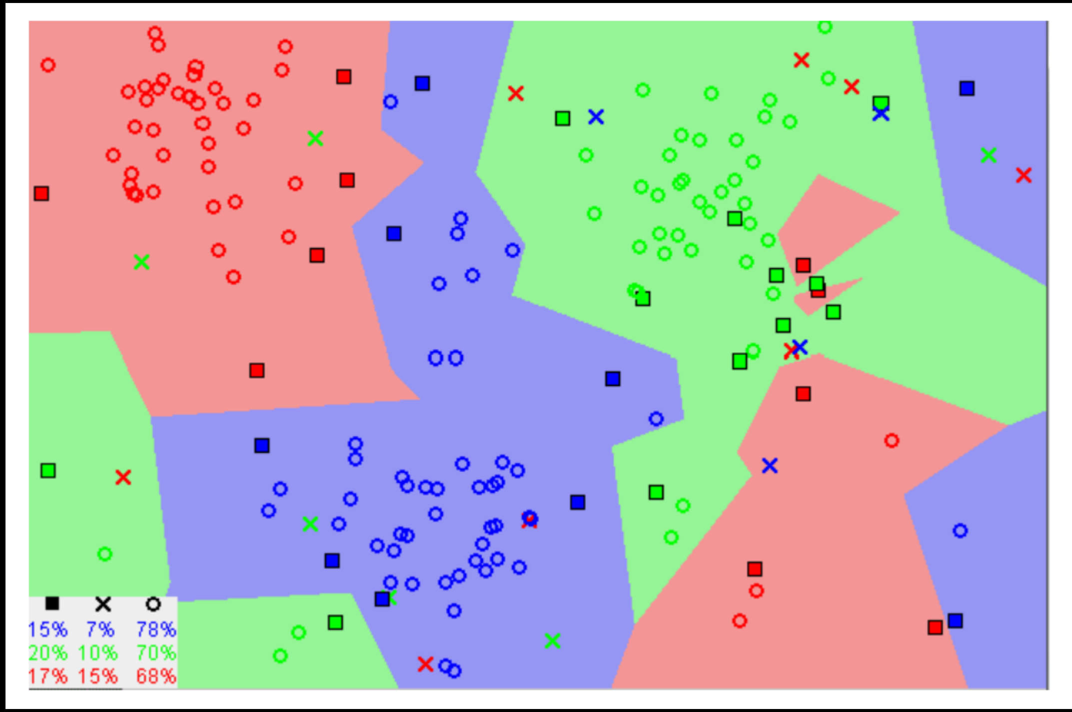
Nine Different Decision Tree Classifiers



Random Forest ensemble for the above Decision Tree classifiers

1.2 K-nearest neighbors (KNN)

<u>Key assumption</u>: similar things exist in close proximity



1.2 K-nearest neighbors (KNN)

How to choosing the right value for K?

- Run KNN algorithm several times with different values of K
- Choose the K that reduces the number of errors we encounter while maintaining the algorithm's ability to predict accurately when it's given new data.

↓ K to 1 = predictions ↓ stable.

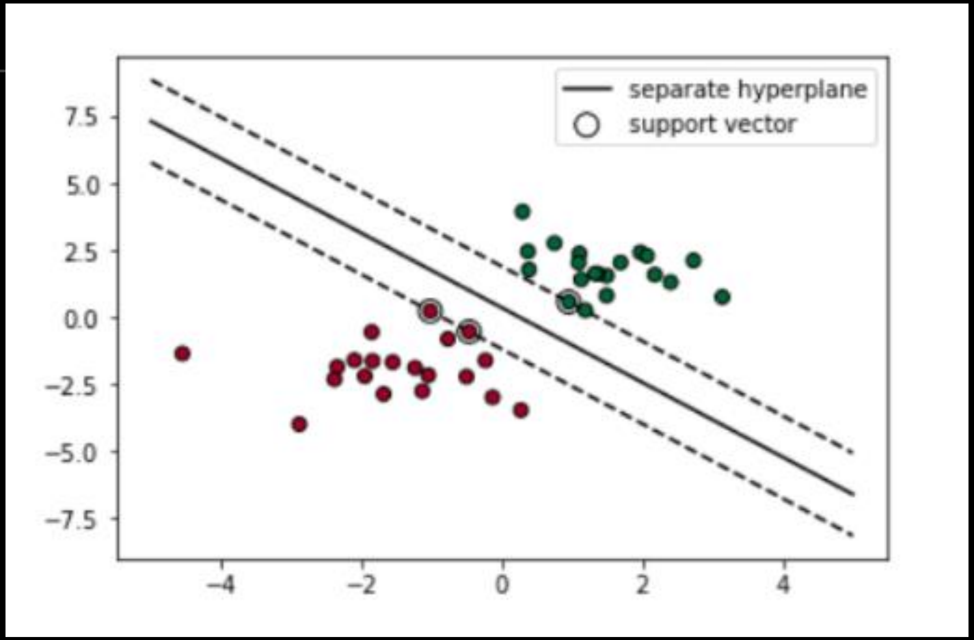
↑ K = predictions ↑ stable (due to majority voting / averaging), BUT @ the cost of increasing number of errors!

NOTE: If you take a majority vote (e.g. picking the mode in a classification problem) among labels, make K an odd number to have a tiebreaker.

1.2 Support Vector Machine (SVM)

Key idea: find the decision boundary to separate different classes and maximize the margin.

Simplest case = linear separable, 2-D



For more details on nonseparable cases & SVM, check

https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496

2. Model Evaluation Methods

For your own self-reading, we will see some in practice ☺

https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

Further home readings?

W7: Check Microsoft Azure ML Algorithm Cheat Sheet & blog posts on how to select the best ML algorithms.

W8: Required articles on word embeddings, word2vec and cosine similarity (also in W7 reading list)

Practice time ☺

Open your Google Colab/ Jupyter Notebook

Practice time ☺

https://github.com/httn21uhh/Text-Analysis-for-Social-Sciences-in-Python

→ W7_supervisedmethods.ipynb + housing_cali.csv

- Download them
- Run the files on your own laptop
- The iPython file is NOT meant for passive scrolling!

This week…

- See some supervised ML methods in action & evaluate the models
- Follow closely the illustrated examples and replicate yourself as the session proceeds.
- Raise your hands to ask questions at any point, including when you think things go too fast/slow for you.