Klasifikacija mina i kamenja iz sonar dataseta

Alen Andrašek, Monika Majstorović, Luka Valenta

Opis problema

- **Dataset: sonar** (UCI repozitorij za SU)
  - 208 podataka (111 mina, 97 kamenja)
  - 60 featurea (vrijednost između 0.0 i 1.0)
  - Svaki od tih 60 brojeva predstavlja energiju unutar određenog frekvencijskog pojasa, integriranu tijekom određenog vremenskog razdoblja.
  - Često korišten dataset za razne edukacijsko-demonstrativne svrha i benchmark novih modela u SU.

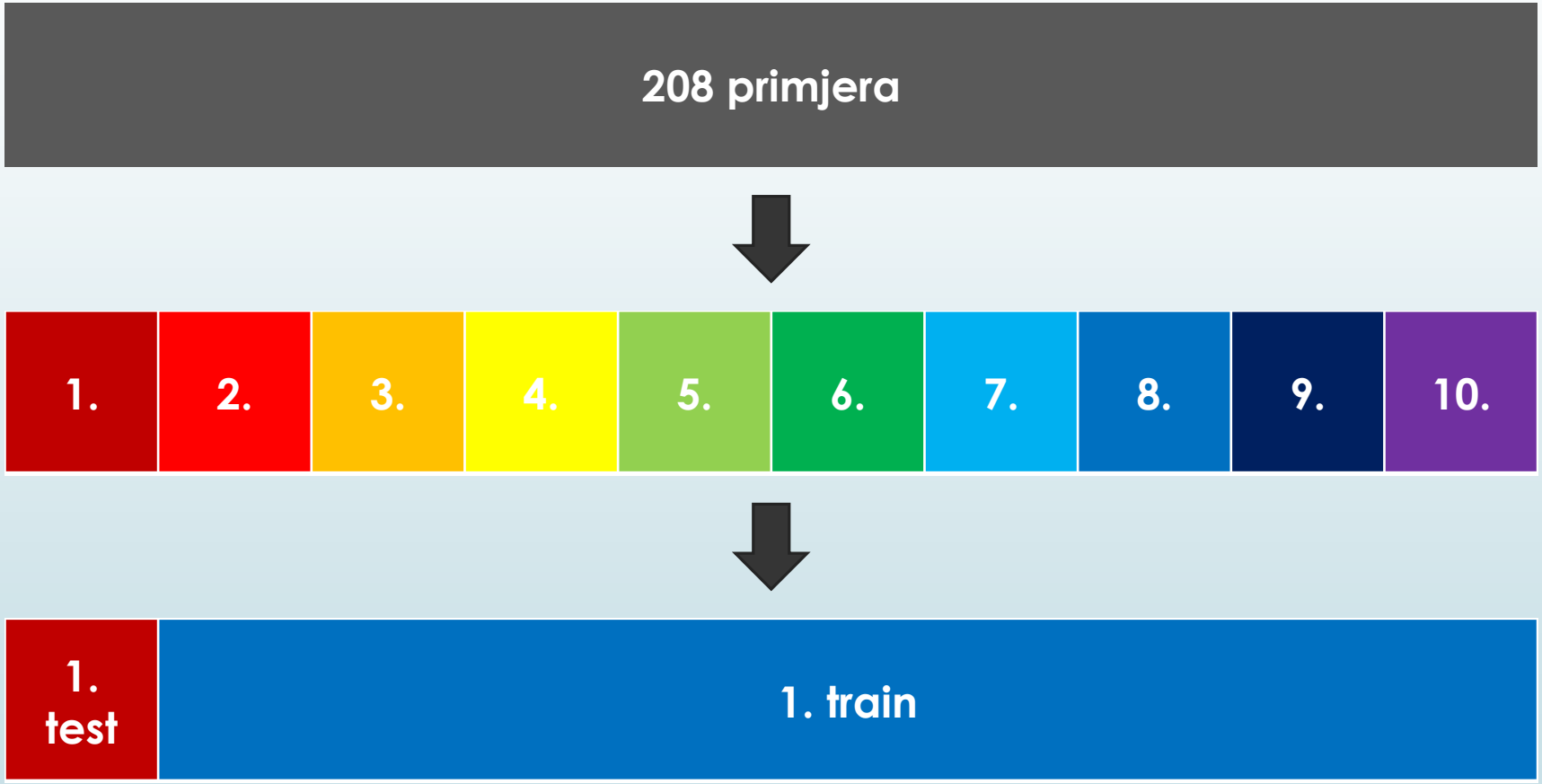
Mali dataset

- Za strojno učenje 208 primjera je vrlo malo. Rad s malim datasetom ima neke prednosti i mane.
- Prednost rada s takvim datasetom je kraće vrijeme izvođenja algoritama na njemu.
- Mane:
  - Mali broj primjera za treniranje i testiranje.
  - Moguća nepouzdanost ocjene uspješnosti različitih algoritama na datasetu zbog premalog skupa za testiranje.
  - Overfitting zbog malog seta za treniranje.
  - Outlieri i šum mogu imati veći utjecaj zbog malog broja primjera.
  - Moguća nejednaka zastupljenost pojedinih klasa u skupovima za treniranje odnosno testiranje.

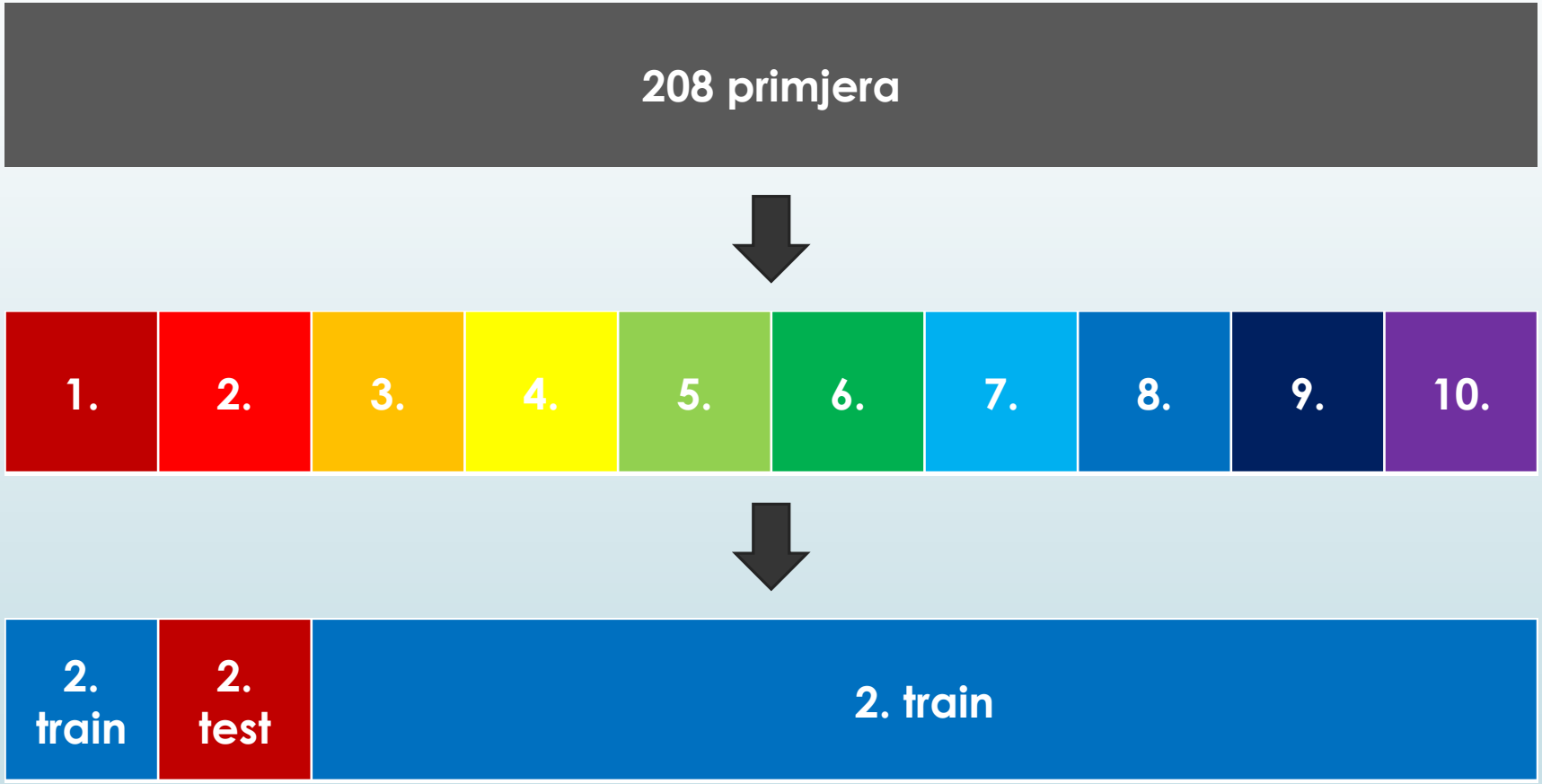
Rješenja problema malog dataseta

- Nalaženje još primjera **nije** opcija.
- Skup primjera dijelimo na **10 disjunktnih podkupova** koji u uniji čine cijeli skup primjera.
- Od njih konstruiramo 10 parova skupova za treniranje i testiranje tako da je svaki od 10 podskupova u jednom paru test set dok svi ostali čine train set.
- **Prednosti ovog postupka:**
  - Sve modele treniramo na oko 90% primjeraka.
  - Svaki primjerak će se točno jednom naći u nekom skupu za testiranje.
  - Uzimanjem prosjeka uspješnosti algoritma na 10 train – test parova dobivamo bolju ocjenu uspješnosti algoritma na datasetu. Smanjuje se ovisnost ocjene o testu za treniranje.
  - Korištenjem stratifikacije pri podjeli na podskupove osiguravamo podjednaku zastupljenost obje klase u skupovima za treniranje odnosno testiranje.

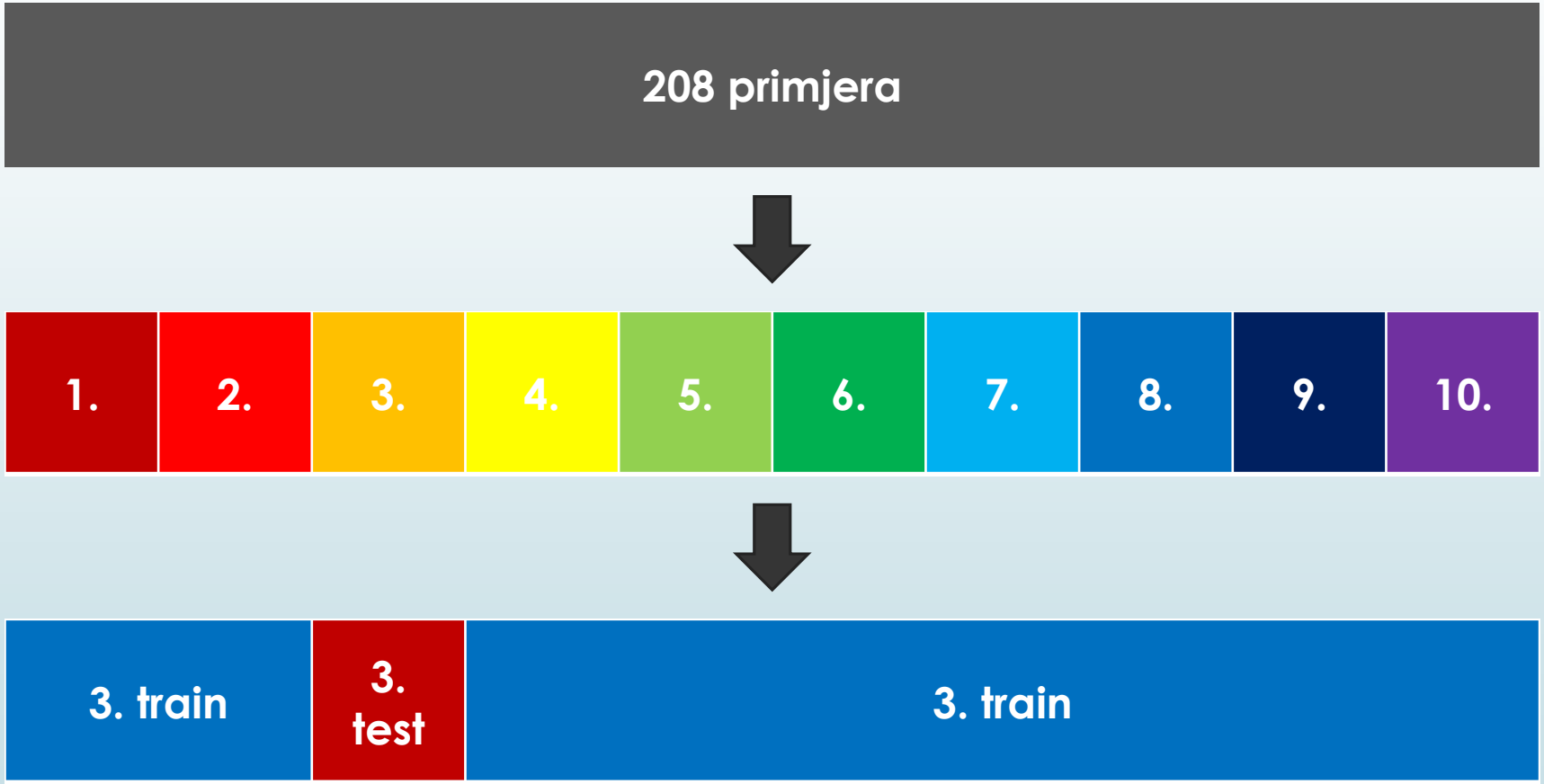
Ilustracija podjele dataseta (1)



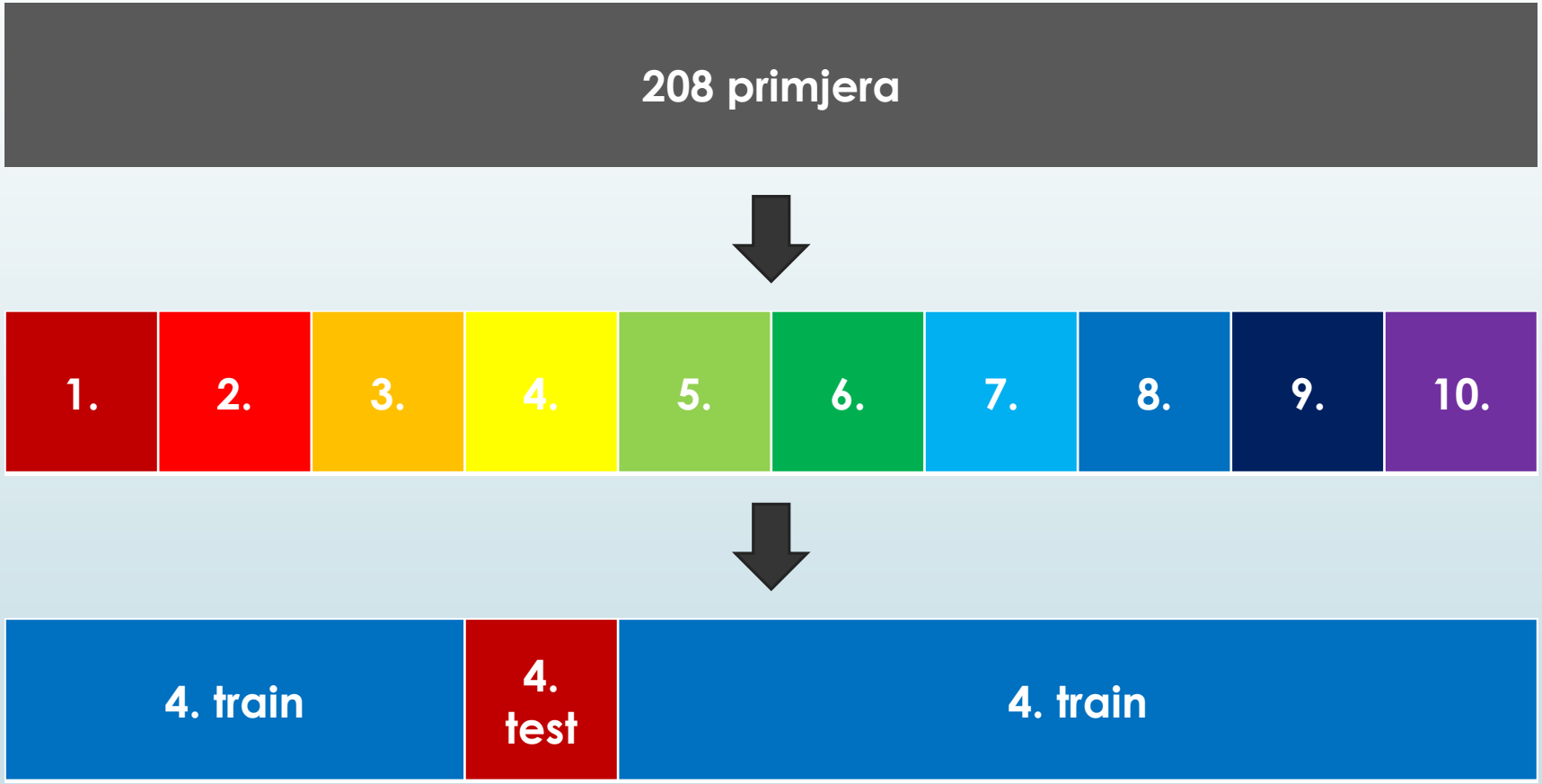
Ilustracija podjele dataseta (2)



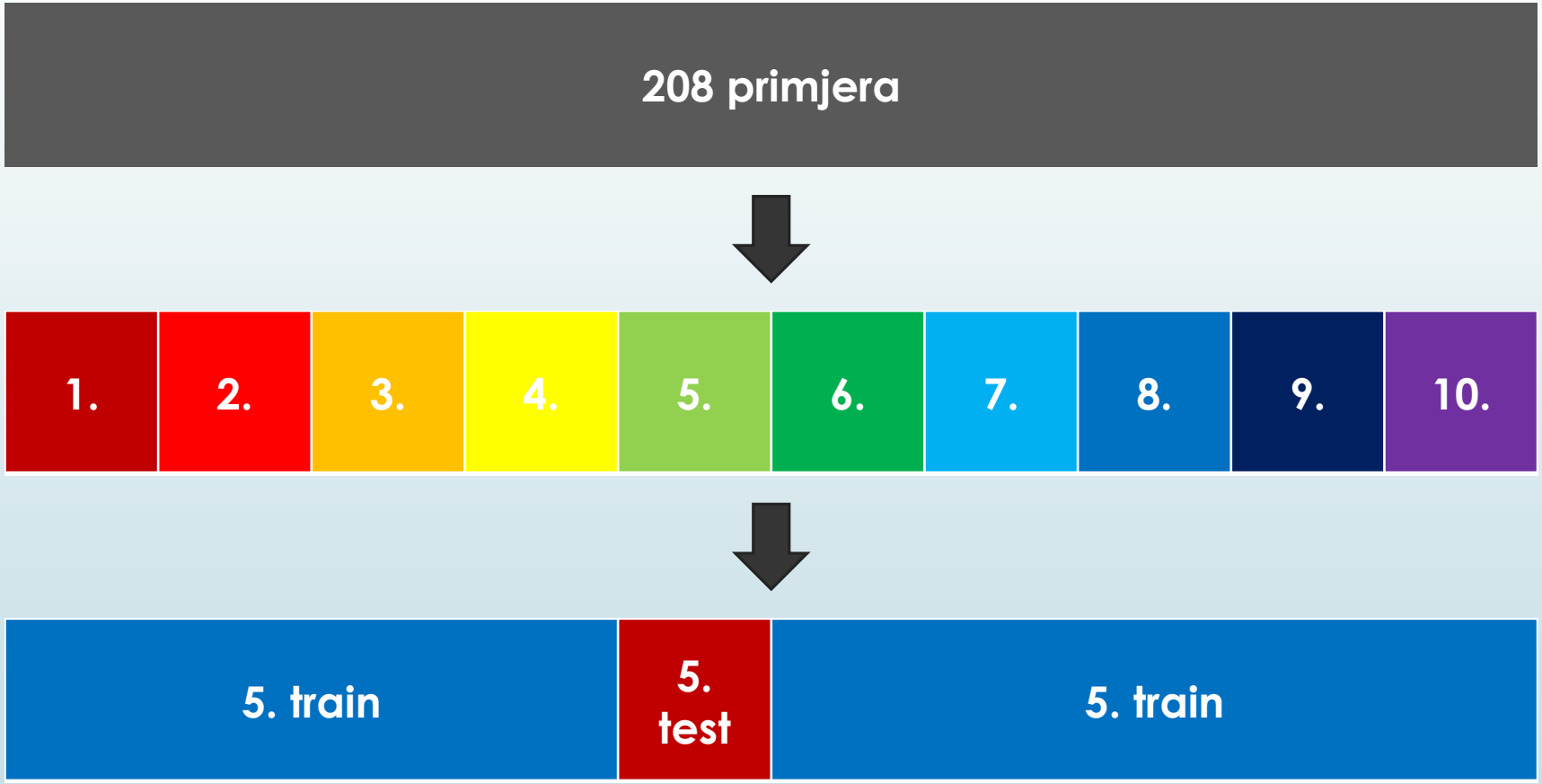
Ilustracija podjele dataseta (3)



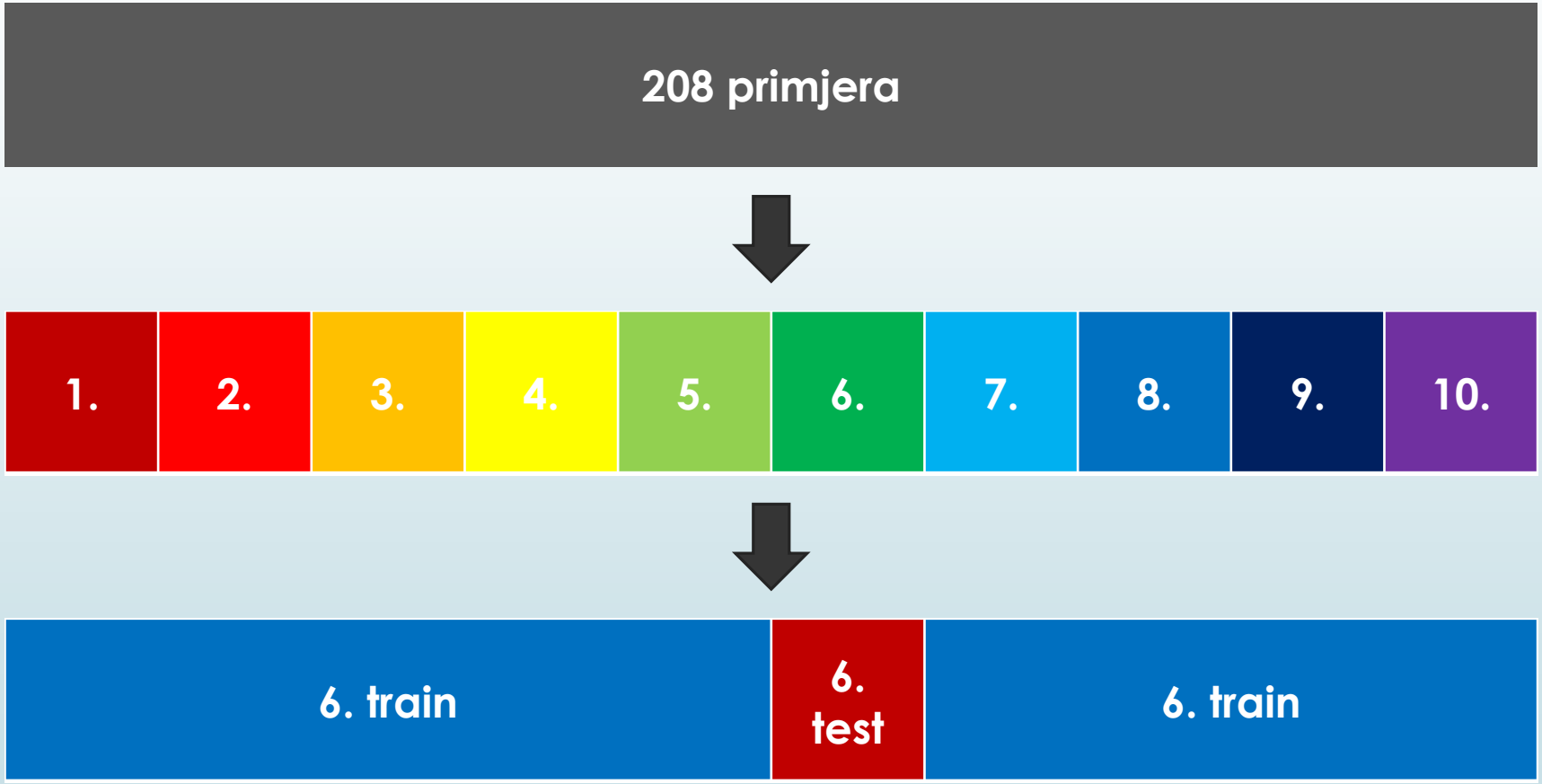
Ilustracija podjele dataseta (4)



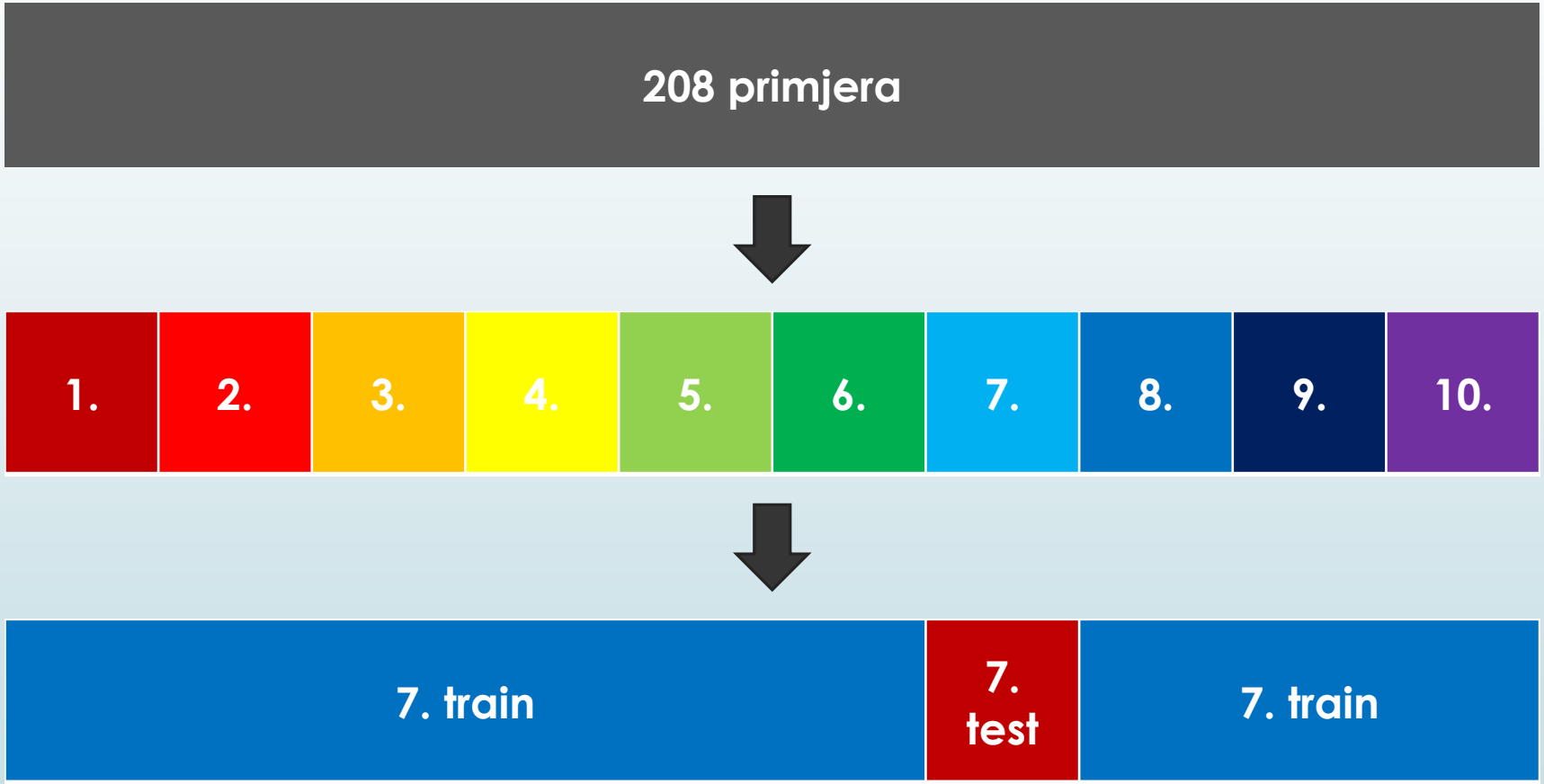
Ilustracija podjele dataseta (5)



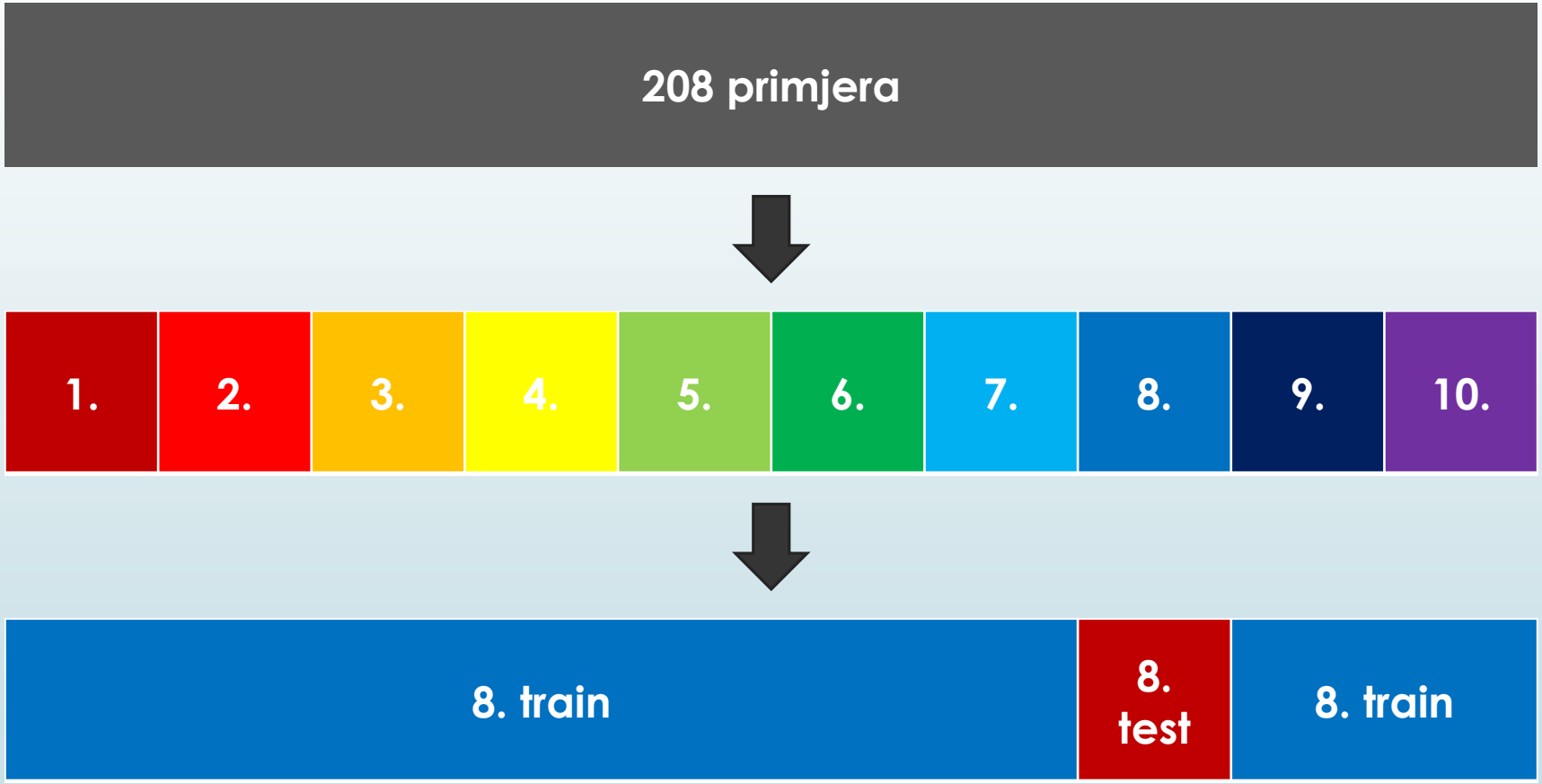
Ilustracija podjele dataseta (6)



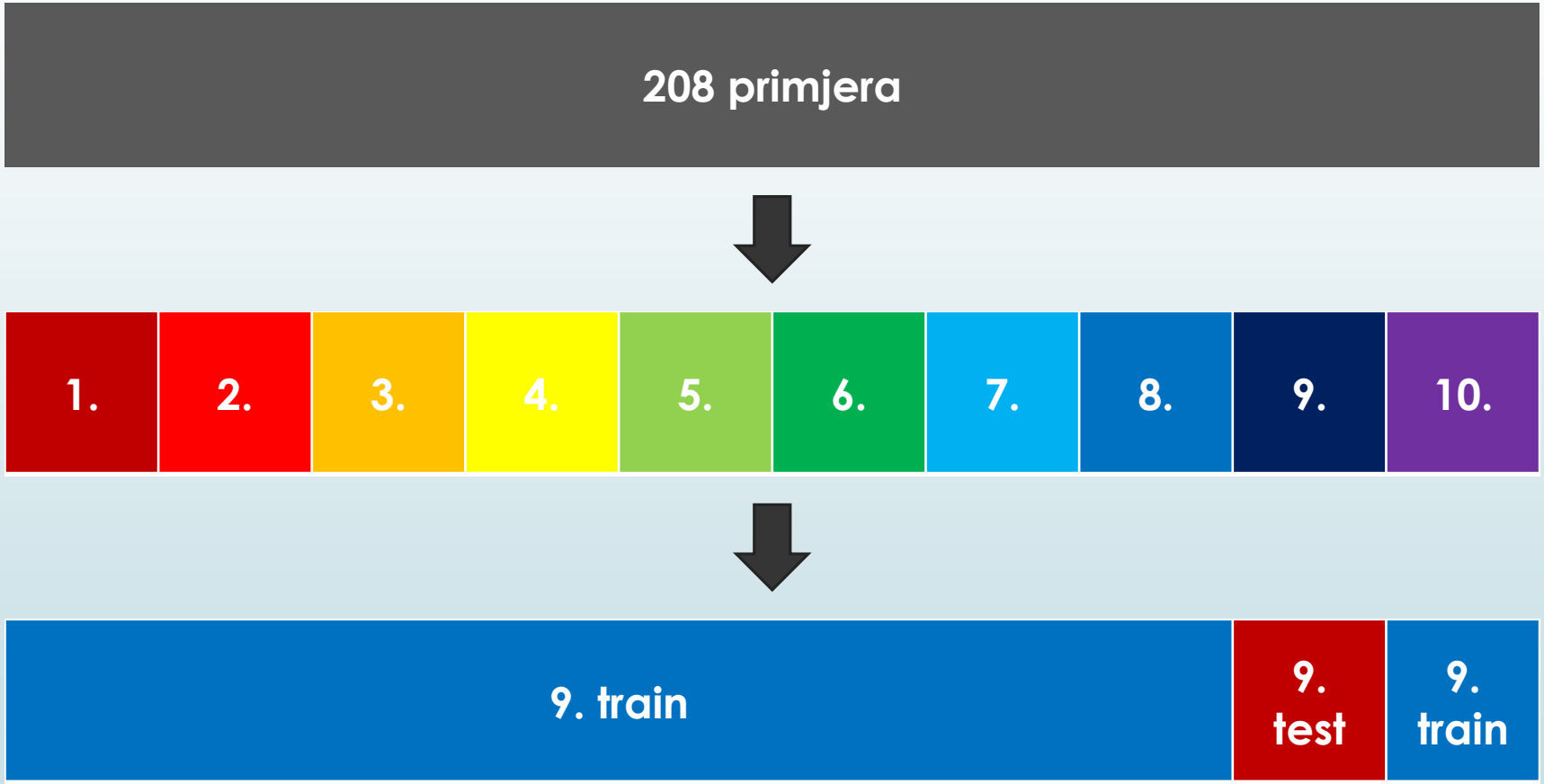
Ilustracija podjele dataseta (7)



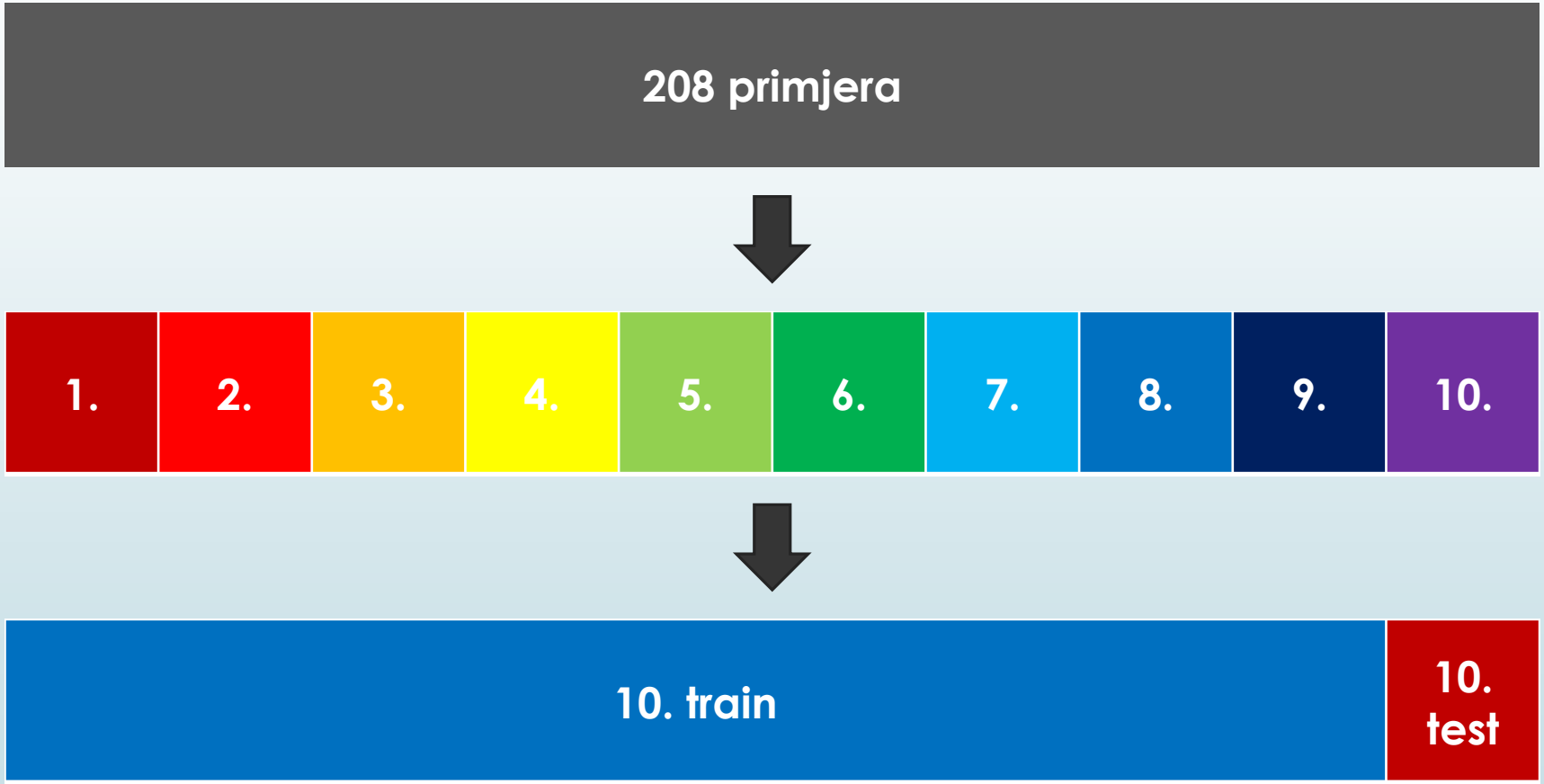
Ilustracija podjele dataseta (8)



Ilustracija podjele dataseta (9)



Ilustracija podjele dataseta (10)



Pretprocesiranje podataka

- Prije pokazane podjele podataka na 10 podskupova na podatcima smo napravili **shuffle** i **PCA**.
- PCA:
  - napravljen na cijelom datasetu
  - zadržavamo 95% varijance
  - konačna dimenzija: 17

SVM

- RBF kernel
- Grid search
  - C dobiven koristeći `numpy.logspace(-1, 3,100)`
  - γ dobiven koristeći `numpy.linspace(0.0001, 10, 100)`
  - Vrijeme izvršavanja: 54 minute
- Randomized search
  - $C \in [1, 1000]$
  - $\gamma \in [0.001, 1]$
  - 100 iteracija
  - Vrijeme izvršavanja: 21 sekunda
- Pipeline + 3-fold cross-validation

K-NN

- Metrike:
  - Euklidska
  - Manhattan
  - Čebiševljeva
- Grid search
  - $k \in \{1, 2, \ldots, 10\}$
  - Vrijeme izvršavanja: 2 sekunde
- Pipeline + 3-fold cross-validation
- Težinski k-NN

Random Forest

- Grid search:
  - Broj stabala $n_{estimators} \in \{1,3, \dots, 19\}$
  - Broj featurea za svako stablo $n_{features} \in \{1,2, \dots 17\}$
- Vrijeme izvršavanja: 79 sekundi
- Pipeline + 3-fold cross-validation

autosklearn

- Automatski odabire algoritme i optimizira hiperparametre.
- Koristi:
  - 15 klasifikatora
  - 14 metoda za pretprocesiranje značajki
  - 4 metode pretprocesiranja podataka
- Konstruira ansamble od modela evaluiranih tijekom optimizacije.
- Kao ulaz su dani **isti parovi** skupova za treniranje i testiranje, ali **bez PCA** na podatcima.
- Vrijeme izvršavanja za svaki par skupova za treniranje i testiranje: **1h**

Rezultati

- Najbolji performans:
  - SVM (Randomized search) s prosječnom točnošću 85.974%
- Najgori performans:
  - Random Forest s prosječnom točnošću 79.1883%

- Hiperparametri:
  - K-NN daje najbolje rezultate na svih 10 test setova upotrebom Čebiševljeve metrike, a broj susjeda varira (k = 1 (5/10), k = 3 (4/10), k = 5 (1/10))

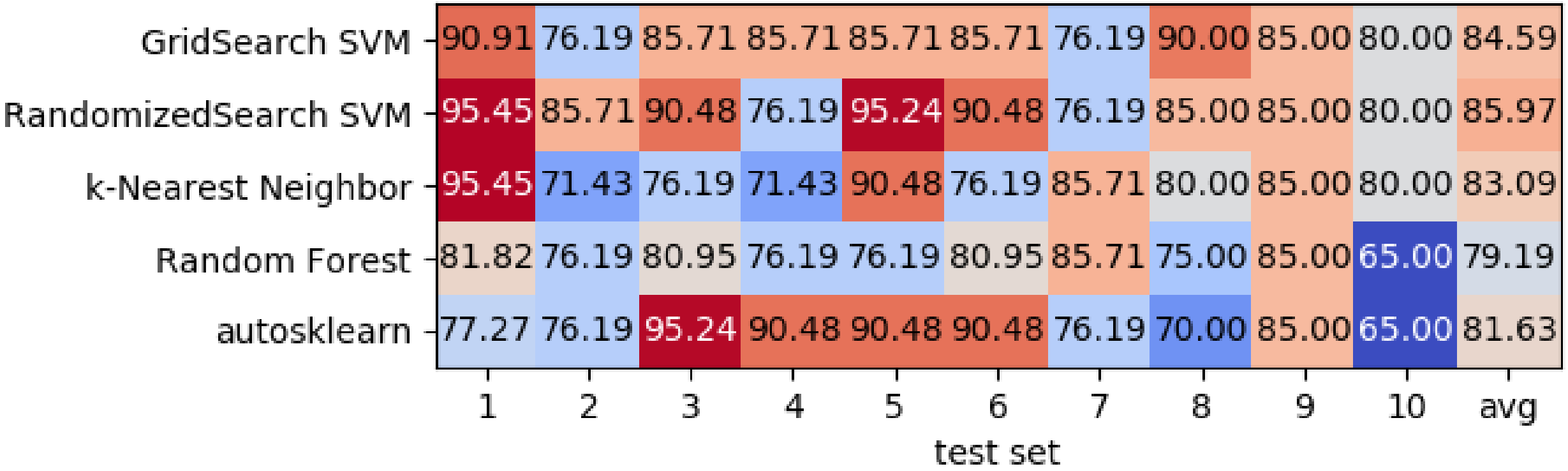
%	SVM (GS)	SVM (RS)	K-NN (Čeb)	RF	autosklearn
AVG točnost	84.5909	85.974	83.0931	79.1883	81.632

Rezultati (usporedba s literaturom)

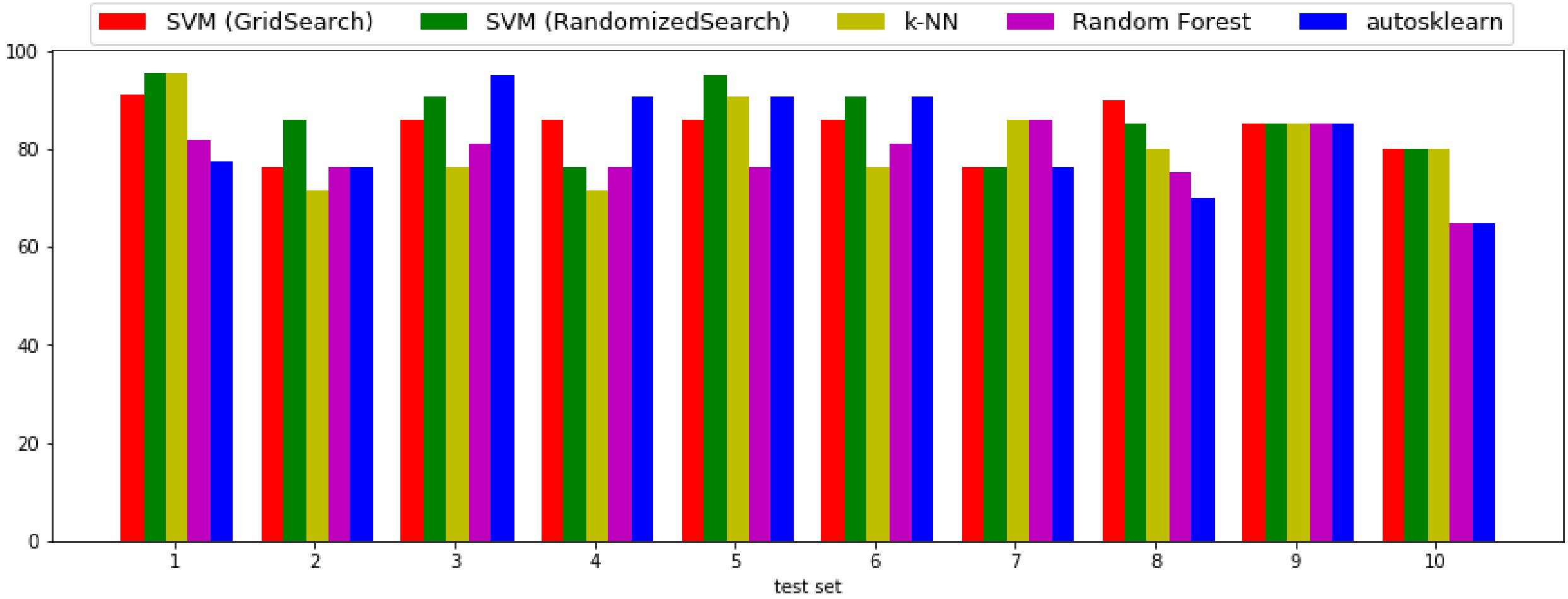
- Autori članka Benchmarking for SVM (2006.) [5] uspoređivali su na raznim datasetovima performanse SVM-a s dotad najboljim performansama mnogih drugih metoda.
- Među ispitanima na sonar setu bili su i k-NN i Random forest.
- Naš SVM pokazao je malo bolji performans nego SVM iz članka, ali bio je lošiji nego k-NN koji je pokazao rezultate bolje od svih modela ispitanih na sonar datasetu.

%	SVM	K-NN	Random Forest
Naši rezultati	85.974	83.0931	79.1883
Rezultati iz [5]	84.56	87.31	83.8

Rezultati


GridSearch SVM
RandomizedSearch SVM
k-Nearest Neighbor
Random Forest
autosklearn
90.91 76.19 85.71 85.71 85.71 85.71 76.19 90.00 85.00 80.00 84.59
95.45 85.71 90.48 76.19 95.24 90.48 76.19 85.00 85.00 80.00 85.97
95.45 71.43 76.19 71.43 90.48 76.19 85.71 80.00 85.00 80.00 83.09
81.82 76.19 80.95 76.19 76.19 80.95 85.71 75.00 85.00 65.00 79.19
77.27 76.19 95.24 90.48 90.48 90.48 76.19 70.00 85.00 65.00 81.63
1 2 3 4 5 6 7 8 9 10 avg
test set

Rezultati – bar chart



Zaključak

- Najbolja točnost: SVM (Randomized Search): 85.97%
- Za metode čiji parametri nisu diskretni, Randomized Search se pokazuje još boljom opcijom za njihovo određivanje nego Grid Search.
- Vidljivo je da izračunata točnost metode ovisi o testnom skupu.
- Uzimanje prosječne točnosti na više skupova za testiranje se pokazala nužnom za stvaranje stvarne slike o točnosti algoritma.

Literatura

- [1] M.R. Mosavi, M. Khishe, A. Ghamgosar; *Classification of Sonar Dataset Using Neural Network Trained by Grey Wolf Optimization*; Iran University of Science and Technology, Teheran; Iran, 2016.
- [2] R.P. Gorman, T. J. Stejnowski; *Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets*; 1987.
- [3] R. K. Jade, L. K. Verma, K. Verma; *Classification using Neural Network and Support Vector Machine for Sonar dataset*; International Journal of Computer Trends and Technology; Vol. 4, Issue 2; pg 116-119; 2013.
- [4] H. T. Hassan, M. U. Khalid, K. Imran; Intelligent Object and Pattern Recognition using Ensembles in Back Propagation Neural Network; International Journal of Electrical & Computer Sciences IJECS-IJENS; Vol. 10, No. 6; pg 52-59; 2010.
- [5] D. Meyer, F. Leisch, K. Hornik; Benchmarking Support Vector Machines; 2002.
- [6] https://medium.com/rants-on-machine-learning/what-to-do-with-small-data d253254d1a89 (Zadnje pristupljeno: 30. travnja 2018.)
- [7] https://pdfs.semanticscholar.org/d6dc/df86df3ece94c2c5effe205d105c561ed5eb.pdf (Zadnje pristupljeno: 30. travnja 2018.)
- [8] https://stats.stackexchange.com/questions/117643/why-use-stratified-cross-validation-why-does-this-not-damage-variance-related-b (Zadnje pristupljeno: 30. travnja 2018.)
- [9] http://cs229.stanford.edu/notes/cs229-notes5.pdf (Zadnje pristupljeno: 30. travnja 2018.)
- [10] J. Bergstra, Y. Bengio; *Random Search for Hyper-Parameter Optimization;* Journal of Machine Learning Research 13, pg. 281-305; 2012.
- [11] Kilian Q. et al., "Distance Metric Learning for Large Margin Nearest Neighbor Classification", Journal of Machine Learning Research 10, pg. 207-244., 2009.
- [12] https://archive.ics.uci.edu/ml/datasets/connectionist+bench+(sonar,+mines+vs.+rocks) (Zadnje pristupljeno: 30. travnja 2018.)
- [13] https://stats.stackexchange.com/questions/61546/optimal-number-of-folds-in-k-fold-cross-validation-is-leave-one-out-cv-always (Zadnje pristupljeno: 30. travnja 2018.)
- [14] M. Feuer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, F. Hutter; Efficient and Robust Automated Machine Learning; NIPS 2015.