# Obrada prirodnih jezika

Elektrotehnički fakultet Univerziteta u Beogradu

Master akademske studije

modul Softversko inženjerstvo

2020/2021

# Obučavanje i evaluacija modela u nadgledanom mašinskom učenju

Vuk Batanović, Elektrotehnički fakultet Univerziteta u Beogradu

# Obučavanje modela

- Pod obučavanjem modela se podrazumeva optimizacija modela korišćenjem raspoloživih parova $(x, y)$
- Cilj obučavanja jeste da se pronađe optimalna hipoteza $h(x)$ – ona koja je najsličnija pravoj funkciji preslikavanja $\mathcal{X} \rightarrow \mathcal{Y}$
- Šta optimizacija tačno znači zavisi od konkretnog algoritma mašinskog učenja
- Skup podataka na osnovu kojih se model optimizuje naziva se skup za obučavanje (engl. *training set*)
- Parametri modela su oni faktori koje je sam algoritam učenja u stanju da optimizuje tokom procesa obučavanja

# Model u mašinskom učenju

- Pod *modelom* se može podrazumevati
  - Algoritam mašinskog učenja (model u širem smislu)
    - Npr. linearna regresija
  - Konkretna funkcija preslikavanja ulaza u izlaz, tj. konkretna hipoteza $h(x)$, dobijena primenom odabranog algoritma mašinskog učenja nad nekim konkretnim skupom podataka (model u užem smislu)
    - Npr. neka konkretna regresiona funkcija

# Evaluacija modela

- Kako znati da li je model A bolji od modela B na određenom zadatku?
- Potrebno je evaluirati modele A i B nad istim skupom podataka i izračunati i uporediti njihove performanse, izražene pomoću odgovarajuće metrike
- Nad kojim skupom podataka je ispravno izvršiti evaluaciju?
  - Da li se za te svrhe može koristiti skup podataka za obučavanje?

# Prilagođenost modela podacima

- Pri obučavanju modela važno je da se izbegne
  - Nedovoljna prilagođenost modela podacima (engl. *underfitting*) – model nije dovoljno iskoristio date podatke za obučavanje
    - Oblik usvojene hipoteze $h(x)$ je isuviše jednostavan u odnosu na stvarnu funkciju preslikavanja $\mathcal{X} \rightarrow \mathcal{Y}$
    - Model nije u stanju da isprati pravilnosti u podacima koje realno postoje
  - Preterana prilagođenost modela podacima (engl. *overfitting*) – model se previše oslonio na date podatke za obučavanje
    - Oblik usvojene hipoteze $h(x)$ je isuviše kompleksan u odnosu na stvarnu funkciju preslikavanja $\mathcal{X} \rightarrow \mathcal{Y}$
    - Model nije u stanju da se ograniči na pravilnosti koje realno postoje u podacima već ih uočava i tamo gde ih realno nema već su proizvod šuma/slučajnosti

# Šum u podacima za obučavanje

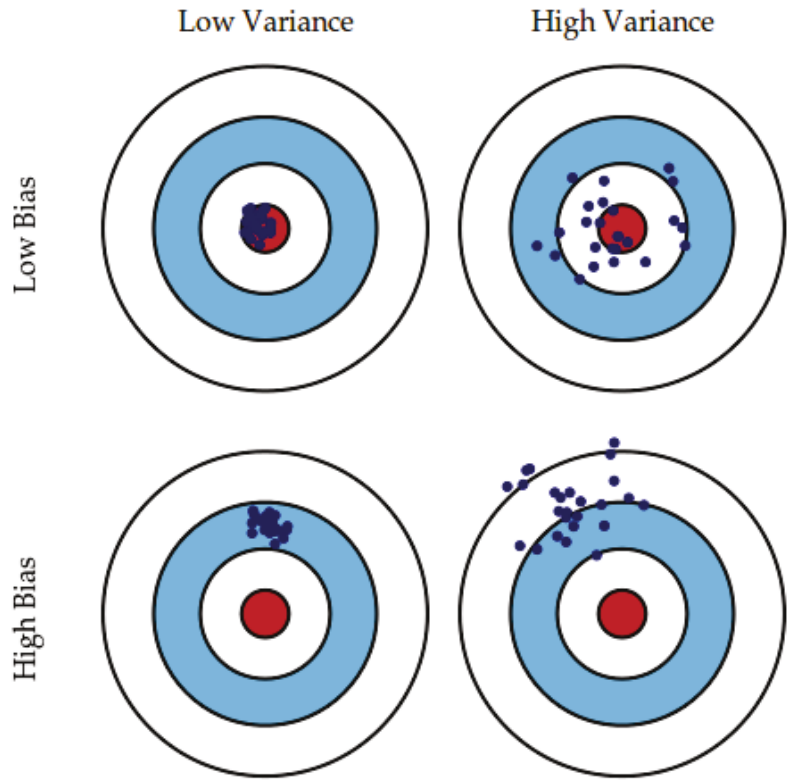
- Pod šumom podrazumevaju se neželjene anomalije u podacima
- Uzroci šuma
  - Nepreciznosti u merenju
  - Greške u unosu podataka
  - Greške u anotaciji podataka
  - Subjektivnost
  - …
- Posledica šuma – pravilnost u podacima tj. signal koji model treba da nauči postaje „zamagljen“
- Model može da počne da se prilagođava šumu tj. da uči šum – preterano prilagođen model

# Prilagođenost modela podacima

- Nedovoljna prilagođenost modela znači da model nije iskoristio podatke da dovoljno (ili čak bilo šta) nauči
- Preterana prilagođenost modela znači da je model prestao da na osnovu podataka uči opšte pravilnosti i počeo da memoriše podatke
- Pronalaženje balansa između nedovoljne i preterane prilagođenosti modela podacima je poznato i kao problem kompromisa između sistematskog odstupanja i varijanse (engl. *bias/variance trade-off*)

# Kompromis između sistematskog odstupanja i varijanse

- Greške zbog sistematskog odstupanja (engl. *bias*) su posledica ograničene fleksibilnosti korišćenog modela mašinskog učenja
  - Nedostatak fleksibilnosti sprečava model da nauči da prepoznaje pravilnosti u podacima za obučavanje
  - Ovakvi modeli su u proseku konzistentni, ali netačni
- Greške zbog varijanse (engl. *variance*) su posledica preterane osetljivosti korišćenog modela mašinskog učenja
  - Prevelika osetljivost modela ga tera da pronalazi pravilnosti i u sitnim, nebitnim varijacijama ulaznih podataka
  - Ovakvi modeli su u proseku tačni, ali nekonzistentni

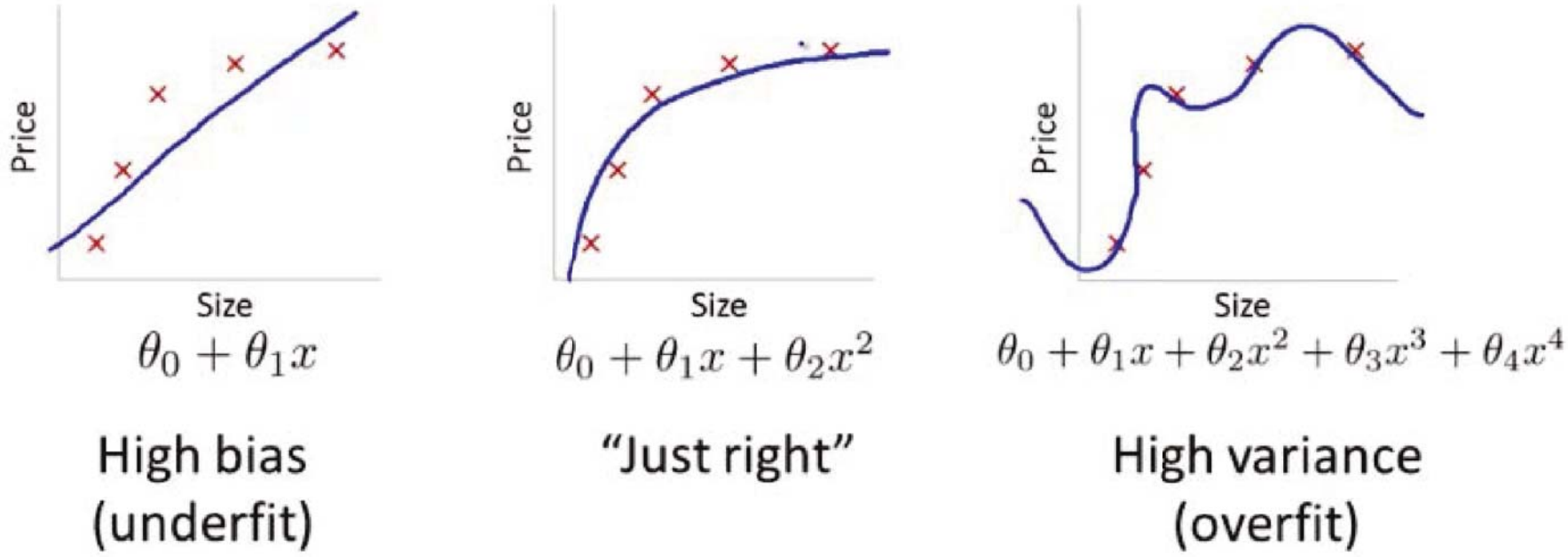



# Ilustracija kompromisa između sistematskog odstupanja i varijanse

Slika preuzeta sa: http://scott.fortmann-roe.com/docs/BiasVariance.html

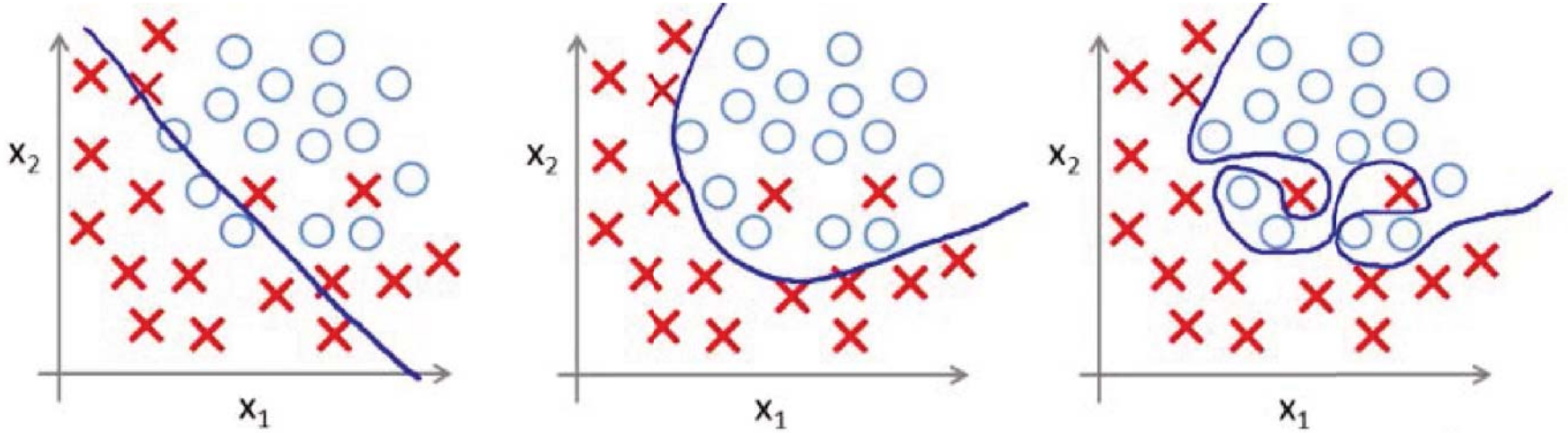
# Zašto kompromis?

- Algoritmi sa malo grešaka zbog varijanse (engl. *low variance*) su oni koji su jednostavniji, ali su zbog jednostavnosti i rigidniji, te skloniji greškama zbog sistematskog odstupanja
  - Linearni modeli
    - Npr. linearna regresija, logistička regresija
- Algoritmi sa malo grešaka zbog sistematskog odstupanja (engl. *low bias*) su oni koji su fleksibilniji, ali su zbog fleksibilnosti i složeniji, te skloniji greškama zbog varijanse
  - Nelinearni modeli
    - Npr. kompleksne arhitekture neuralnih mreža




## Ilustracija efekta nedovoljne i preterane prilagođenosti modela podacima na problemu regresije

Slika preuzeta sa: Andrew Ng, Machine Learning, Coursera




$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

( $g$ = sigmoid function)

**UNDERFITTING**
**(high bias)**

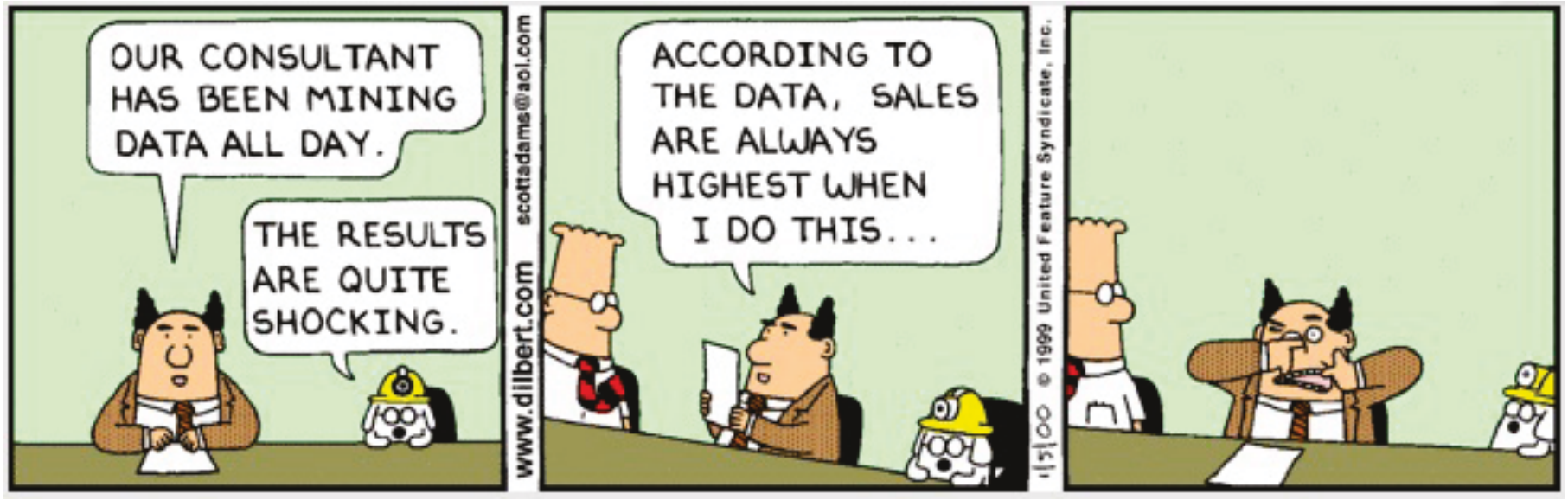
$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots$

**OVERFITTING**
**(high variance)**

## Ilustracija efekta nedovoljne i preterane prilagođenosti modela podacima na problemu klasifikacije

Slika preuzeta sa: Andrew Ng, Machine Learning, Coursera




# Ilustracija efekta preterane prilagođenosti modela u životu

Slika preuzeta sa: http://dilbert.com/

# Evaluacija modela na skupu za testiranje

- Obučeni model je samim obučavanjem prilagođen podacima za obučavanje
  - Možda je i preterano prilagođen
- Ako bi se evaluacija radila nad istim podacima, složeniji/fleksibilniji model bi uvek bio bolji, jer je sposobniji da se prilagodi podacima od rigidnijeg modela
  - To samo znači da složeniji model bolje modelira šum, a ne željene pravilnosti
- Da bi se dobila objektivna, nepristrasna procena performansi modela, evaluaciju je neophodno sprovesti nad novim, do tog trenutka nedirnutim skupom podataka – time se procenjuje sposobnost generalizacije kreiranog modela
  - Takav skup se naziv skupom za testiranje (engl. *test set*)

# Evaluacija modela na skupu za testiranje

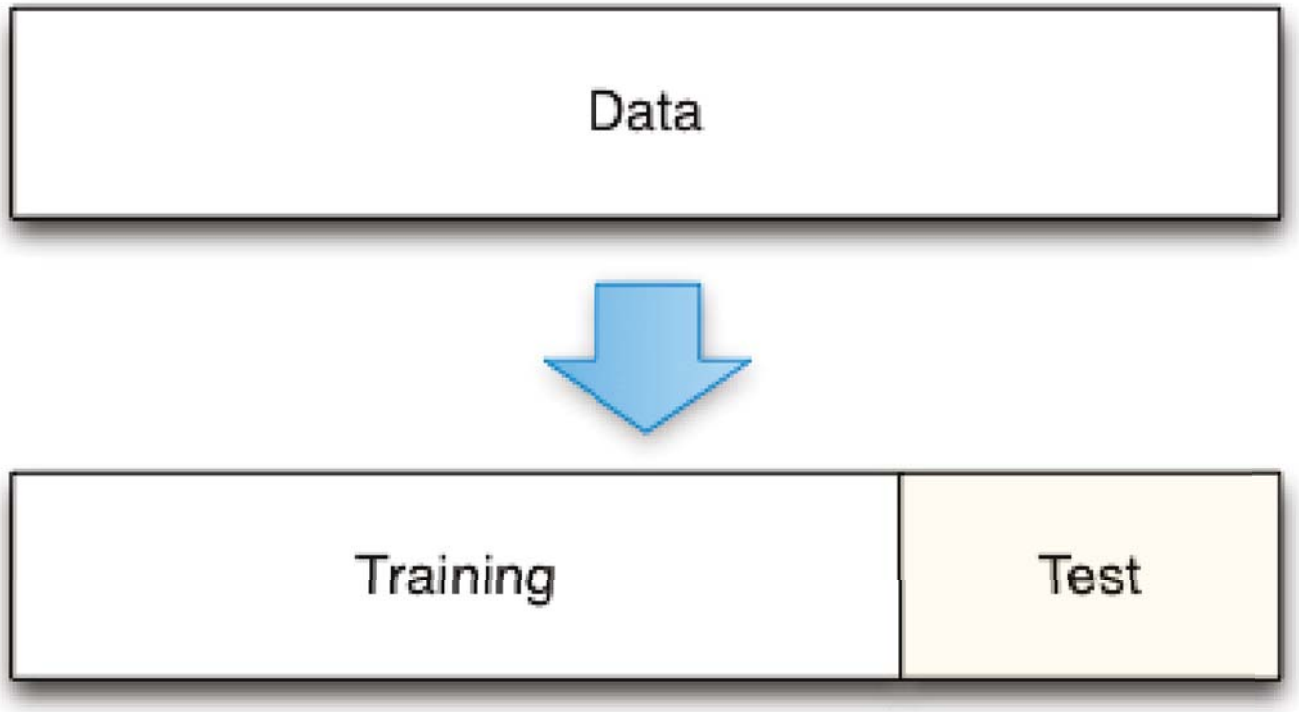
- Obično se dešava da se postojeći skup podataka na početku rada sa modelom mašinskog učenja nasumično podeli na dva – skup za obučavanje i skup za testiranje
- Koliki deo podataka treba da bude u kom skupu?
  - Često se koristi podela 70% / 30% – 70% podataka za obučavanje, 30% za testiranje

# Evaluacija modela na skupu za testiranje

- Važno je da i podaci za obučavanje i podaci za testiranje budu odabrani iz iste opšte populacije podataka
  - Drugim rečima, da podaci iz oba skupa imaju slične karakteristike
- Da bi se ovo postiglo često se primenjuje *stratifikacija*
  - Stratifikacija znači da se pri podeli podataka na skup za obučavanje i skup za testiranje obezbedi da oba skupa imaju istu raspodelu podataka
  - Ovo je dosta teško uraditi za složenije probleme te se koristi jednostavnija varijanta stratifikacije gde se obezbeđuje ista raspodela izlazne vrednosti $y$
    - Klasifikacija – ista frekventnost svih klasa u oba skupa podataka
    - Regresija – (približno) ista frekventnost svih izlaznih vrednosti u oba skupa podataka

# Evaluacija modela na skupu za testiranje

- Ako nije došlo do preterane prilagođenosti modela, performanse na skupu za testiranje su obično slične ili blago niže od onih na skupu za obučavanje
- Ako je došlo do preterane prilagođenosti modela, performanse na skupu za testiranje su primetno lošije od onih na skupu za obučavanje
  - Preterano prilagođen model ima veoma slabu sposobnost generalizacije




## Ilustracija korišćenja odvojenog skupa za testiranje (engl. *holdout test set*)

Slika preuzeta sa: http://scott.fortmann-roe.com/docs/MeasuringError.html

# Kako izbeći nedovoljnu prilagođenost modela?

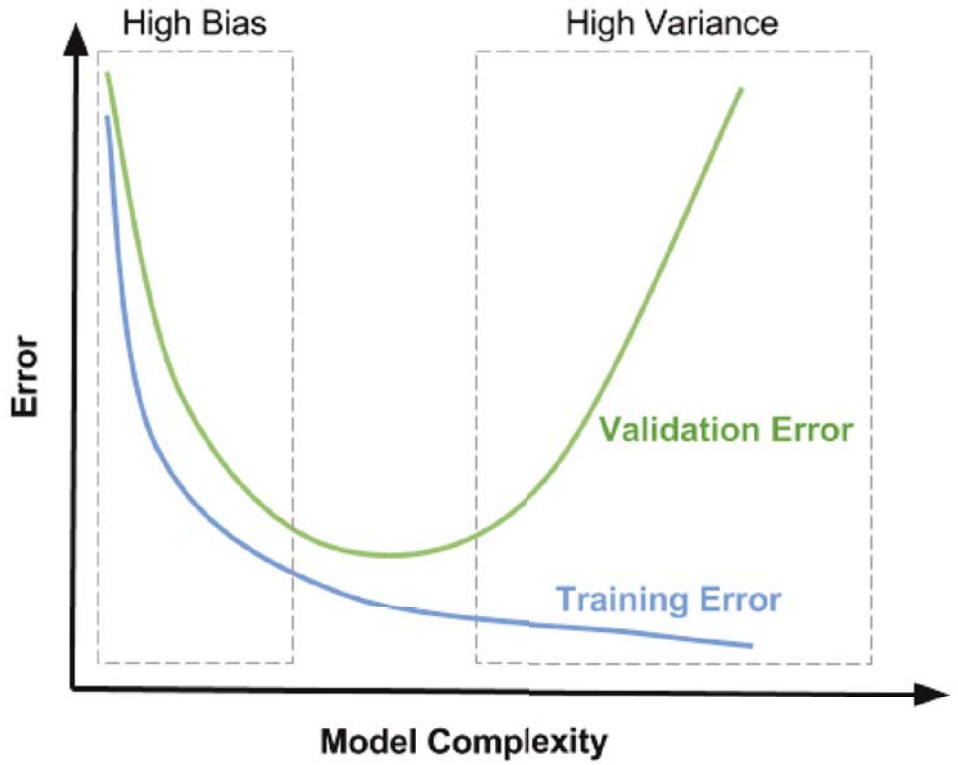
- Odabrati dovoljno fleksibilan algoritam mašinskog učenja za zadati problem
- Odabrati pogodne aspekte ulaznih podataka za njihove odlike/atribute koje se koriste pri obučavanju modela
- Pružiti modelu dovoljan broj kvalitetno odabranih odlika

# Kako izbeći preteranu prilagođenost modela?

- Izbeći korišćenje nepotrebno složenih algoritama za posmatrani problem
- Koristiti neki poseban metod za kontrolisanje složenosti modela (npr. regularizaciju)
- Povećati broj podataka u skupu za obučavanje
- Eliminisati suvišne odlike iz modela (kod nekih algoritama nepotrebno)
- Ukoliko je obučavanje iterativan proces, zaustaviti ga u trenutku kada model počne da se preterano prilagođava
  - Da bi se ispravno procenilo u kom trenutku do ovoga dolazi, neophodno je pratiti performanse modela ne samo nad podacima za obučavanje nego i još nekom skupu podataka – skupu za validaciju (engl. *validation set*)

# Validacija modela

- Pod greškom modela nad nekim podacima $(x, y)$ podrazumeva se odstupanje između izlaza modela za te podatke i njima pridruženih tačnih odgovora
- Različiti algoritmi mašinskog učenja koriste različite oblike funkcije greške
- Ako je obučavanje iterativan proces, prilikom obučavanja greška modela nad podacima za obučavanje se kontinualno smanjuje
- Međutim, greška modela nad nekim drugim skupom podataka iste prirode opada tokom obučavanja samo donekle, a onda počinje da raste
  - To je signal da je model ušao u zonu preterane prilagođenosti podacima za obučavanje

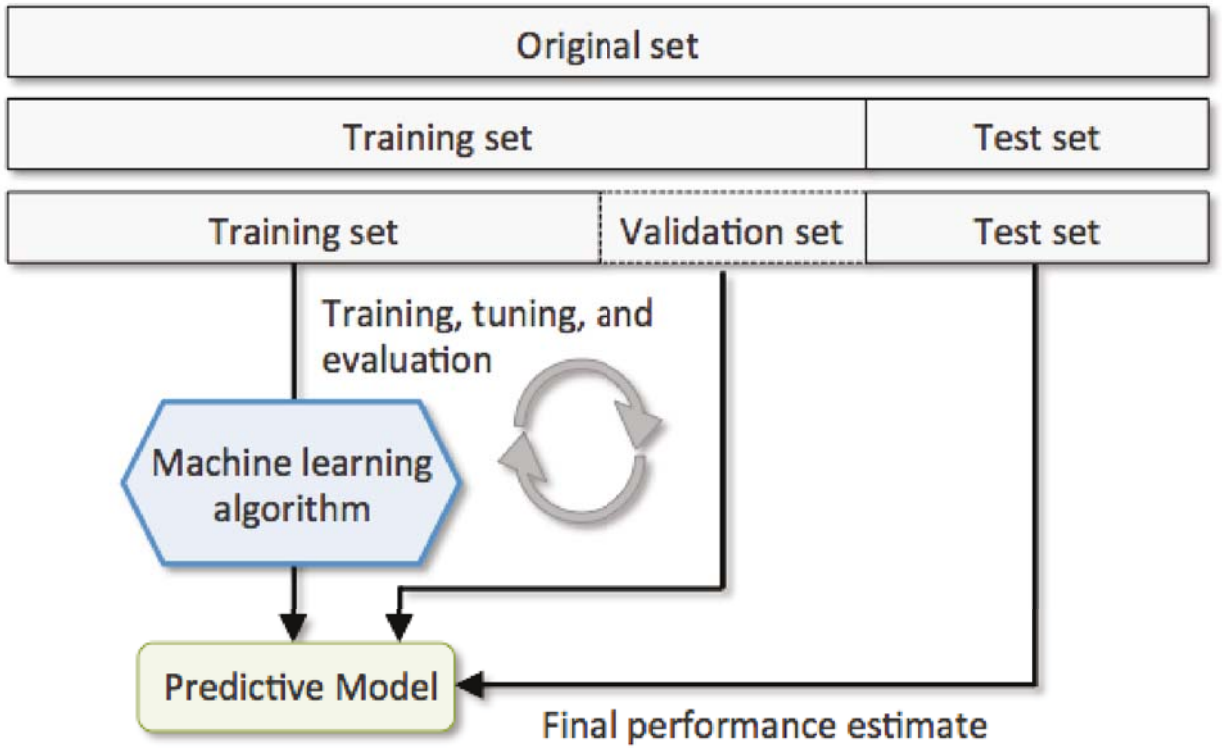



# Ilustracija primene odvojenog skupa za validaciju radi izbegavanja preterane prilagođenosti modela podacima

Slika preuzeta sa: http://www.luigifreda.com/2017/03/22/bias-variance-tradeoff/

# Validacija modela

- Skup podataka za validaciju mora da bude odvojen od skupa za obučavanje i skupa za testiranje
  - Ako bi se skup za testiranje koristio za validaciju, time bi on postao deo optimizacije modela
  - Evaluacija sprovedena nad tako upotrebljenim skupom za testiranje ne bi više bila objektivna i nepristrasna – rezultati bi bili bolji od realnih
- Koliki deo podataka treba da bude u kom skupu?
  - Ne postoji čvrsto pravilo
  - Često se koriste podele 50% - 25% - 25% ili 60% - 20% - 20%
- I za kreiranje validacionog skupa važna je stratifikovana podela
- Skup za validaciju se ponekad naziva i skupom za razvoj (engl. *development set / dev set)*




# Ilustracija korišćenja odvojenog skupa podataka za validaciju i za testiranje

Slika preuzeta sa: http://www.cs.nthu.edu.tw/~shwu/courses/ml/labs/08_CV_Ensembling/08_CV_Ensembling.html

# Hiperparametri i njihova optimizacija

- Parametri modela – faktori ponašanja modela koje sam algoritam učenja optimizuje tokom procesa obučavanja
- Pored njih, ponašanje modela može zavisiti i od određenih faktora koje algoritam učenja nije u stanju da optimizuje
  - Takvi faktori se nazivaju *hiperparametrima* modela
- Vrednosti hiperparametara se moraju zadati ručno, pre početka obučavanja modela

# Hiperparametri i njihova optimizacija

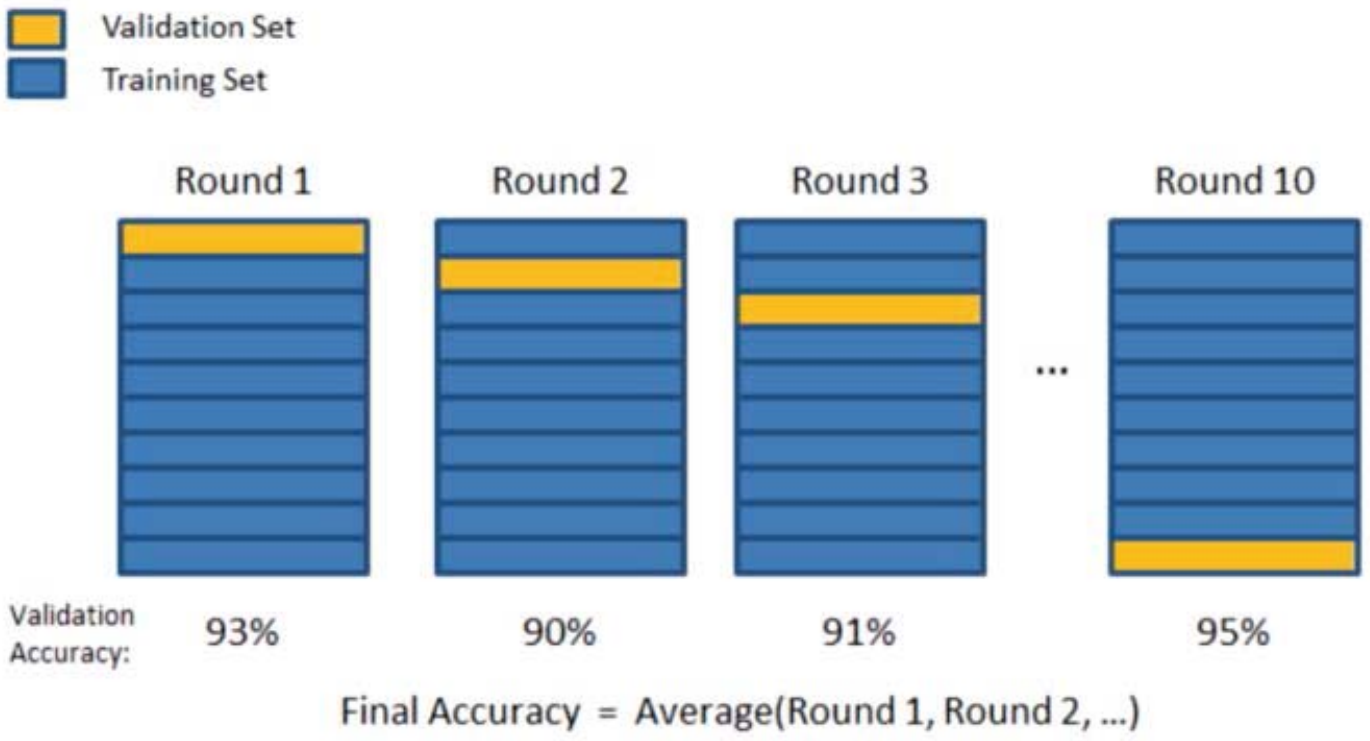
- Optimizacija hiperparametara se vrši istraživanjem različitih kombinacija njihovih vrednosti
- Najjednostavniji pristup je pretraga po mreži (engl. *grid search*), gde se sistematski ispituje svaka moguća kombinacija vrednosti hiperparametara
- Da bi evaluacija modela ostala nepristrasna, hiperparametri se moraju optimizovati pomoću validacije modela – usvaja se onaj set hiperparametara koji na skupu za validaciju daje najbolje rezultate
- Optimizacija vrednosti hiperparametara se takođe naziva i odabirom modela (engl. *model selection*), jer se odabirom njihovih vrednosti vrši i odabir jedne određene vrste modela iz šire familije modela (npr. kvadratne funkcije iz familije polinomijalnih funkcija)

# Unakrsna validacija

- Validacija modela pomoću odvojenog skupa podataka za validaciju ima dva nedostatka
  - Ovim pristupom se efektivno smanjuje količina podataka koje model koristi pri učenju
    - Samim tim, osim ako je umanjeni skup za obučavanje ionako izuzetno veliki, realno je očekivati nešto slabije performanse modela nego što bi bile pri obučavanju sa celokupnim skupom podataka za obučavanje
  - Optimalni hiperparametri mogu da dosta variraju u zavisnosti od nasumične podele podataka na deo za obučavanje i deo za validaciju
    - Što manje podataka je na raspolaganju, to je ovaj efekat primetniji
- Alternativno rešenje za validaciju modela je unakrsna validacija (engl. *cross-validation*)

# Unakrsna validacija

- Kod unakrsne validacije početni skup podataka (sa izuzetkom skupa za testiranje) se deli na $k$ delova, koji se nazivaju *slojevima* (engl. *k-fold cross-validation*)
- Validacija se sprovodi u $k$ prolaza
  - U prolazu $i$ model se validira na $i$-tom sloju, a obučava na svim ostalim zajedno
  - Finalni rezultati validacije se dobijaju uprosečavanjem rezultata na pojedinačnim slojevima
- Najčešće se koristi stratifikovana unakrsna validacija sa 3, 5 ili 10 slojeva
  - Više slojeva – celokupan proces obučavanja duže traje; manja varijansa rezultata
  - Manje slojeva – manji % podataka se koristi za obučavanje u svakom prolazu; veća varijansa rezultata
- Na kraju se model testira na posebnom skupu podataka za testiranje

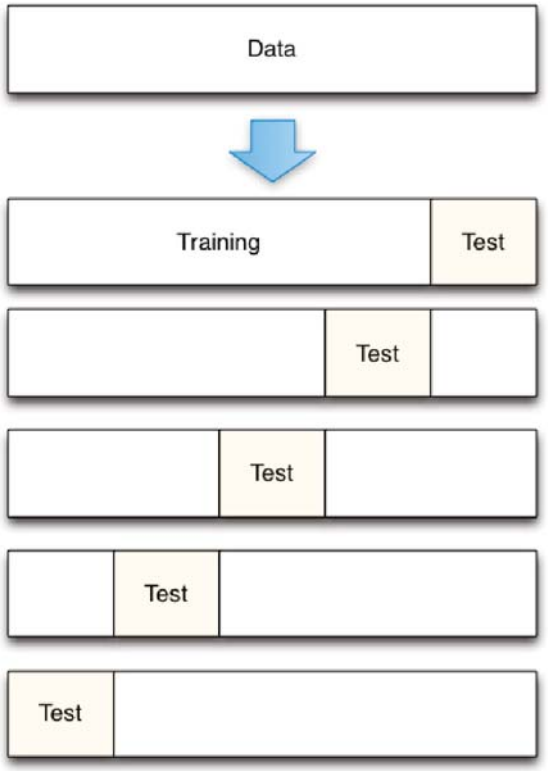



# Ilustracija unakrsne validacije

Slika preuzeta sa: http://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6

# Unakrsna validacija za evaluaciju modela

- Postupak unakrsne validacije se može koristiti i za evaluaciju modela
- Terminološka zabuna – treba razlikovati *validaciju* kao deo procesa optimizacije modela od *unakrsne validacije* kao metode sprovođenja validacije i/ili evaluacije
- Evaluacija pomoću izdvojenog skupa podataka za testiranje ima slične nedostatke kao i validacija pomoću izdvojenog skupa za validaciju
  - Smanjivanje količine podataka koje model koristi pri obučavanju
  - Velika varijansa dobijenih rezultata evaluacije jer zavise od samo jedne nasumične podele početnog skupa na deo za obučavanje i deo za testiranje
    - Što manje podataka je na raspolaganju, to je ovaj efekat primetniji
- Unakrsnom validacijom se svi podaci koriste za evaluaciju, pa je procena performansi pouzdanija

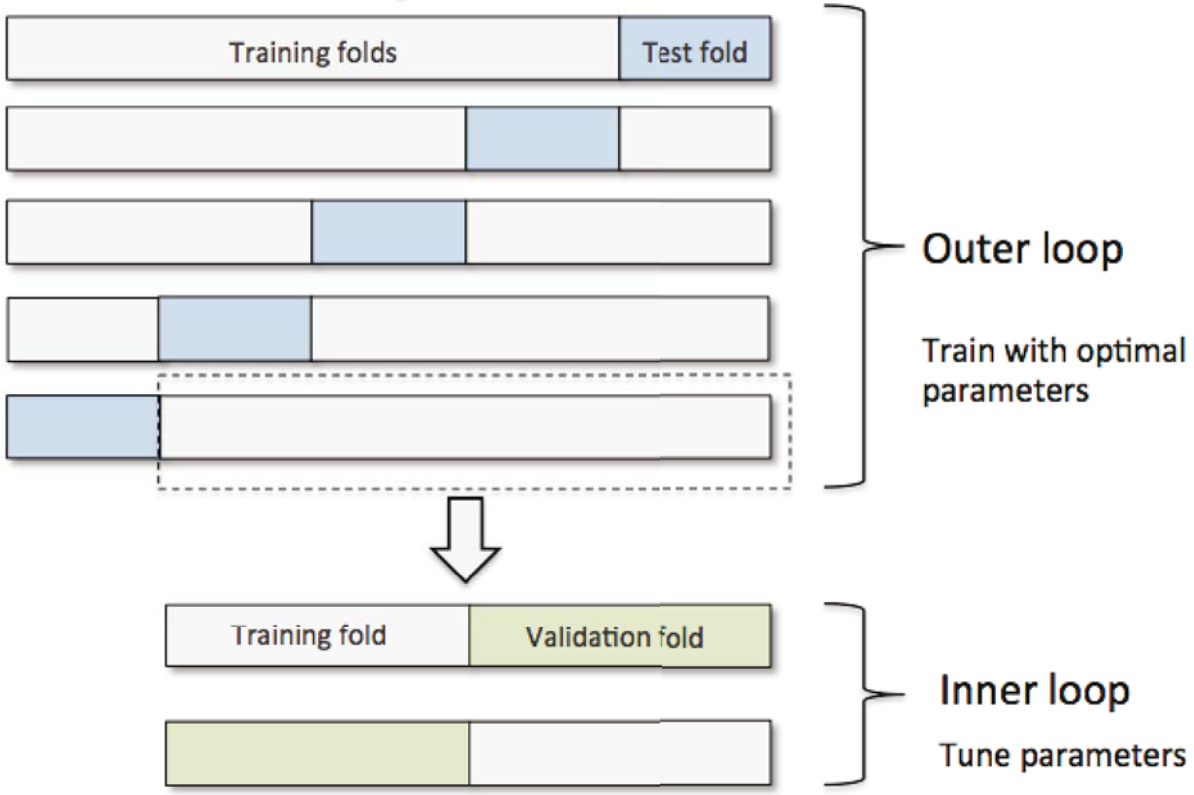



# Ilustracija evaluacije modela pomoću unakrsne validacije

Slika preuzeta sa: http://scott.fortmann-roe.com/docs/MeasuringError.html

# Ugnežđena unakrsna validacija

- Ukoliko je potrebna i validacija i evaluacija modela, onda se validacija sprovodi u ugnežđenoj unakrsnoj validaciji, dok se evaluacija modela sprovodi u spoljnjoj
- Svi slojevi za obučavanje u spoljnoj unakrsnoj validaciji se tretiraju kao jedan skup koji se zatim deli na nove slojeve u ugnežđenoj unakrsnoj validaciji
- Broj slojeva u spoljnoj i u unutrašnjoj/ugnežđenoj unakrsnoj validaciji se može proizvoljno razlikovati
  - Neretko se za ugnežđenu unakrsnu validaciju koristi manji broj slojeva, zbog povećanja brzine obučavanja




# Ilustracija ugnežđene unakrsne validacije

Slika preuzeta sa: http://sebastianraschka.com/faq/docs/evaluate-a-model.html

# Metrike za merenje performansi u regresiji

- $h\left(x^{(i)}\right)$ je vrednost hipoteze za podatak $i$, $y^{(i)}$ je data tačna izlazna vrednost za podatak $i$, $m$ je broj podataka koji se koriste za testiranje
- RMSE (engl. *Root Mean Squared Error*) – koren srednje kvadratne greške - najrašireniji metrika u regresiji:

$$RMSE = \sqrt[2]{\frac{\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2}{m}}$$

- MAE (engl. *Mean Absolute Error*) – srednja apsolutna greška:

$$MAE = \frac{\sum_{i=1}^{m}\left|h\left(x^{(i)}\right) - y^{(i)}\right|}{m}$$

# Metrike za merenje performansi u regresiji

- RRSE (engl. *Root Relative Squared Error*) – koren relativne kvadratne greške

$$RRSE = \sqrt[2]{\frac{\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2}{\sum_{i=1}^{m}(\bar{y} - y^{(i)})^2}}$$

- gde je $\bar{y}$ srednja vrednost izlazne promenljive u skupu za obučavanje
- RAE (engl. *Relative Absolute Error*) – relativna apsolutna greška

$$RAE = \frac{\sum_{i=1}^{m}\left|h\left(x^{(i)}\right) - y^{(i)}\right|}{\sum_{i=1}^{m}|\bar{y} - y^{(i)}|}$$

# Metrike za merenje performansi u regresiji

- Za sve navedene metrike važi da su niže vrednosti bolje
- RMSE i MAE
  - Izražavaju se u istim jedinicama kao izlazna promenljiva $y$
  - Posebno korisni ako se zna kolika greška je prihvatljiva u posmatranom domenu
- RRSE i RAE
  - Relativne metrike tako da nemaju jedinicu
  - Pogodniji za poređenje performansi modela između više različitih domena

# Metrike za merenje performansi u binarnoj klasifikaciji

- Matrica zabune (engl. *Confusion Matrix*) – matrica u kojoj je predstavljena raspodela podataka u zavisnosti od njihove stvarne klase i one u koju ih je model svrstao
- Pošto se radi o binarnoj klasifikaciji, iz konvencije jedna klasa se označava kao pozitivna a druga kao negativna
- Koriste se sledeće oznake:
  - *True Positives* (TP) – ispravno klasifikovani podaci pozitivne klase
  - *True Negatives* (TN) – ispravno klasifikovani podaci negativne klase
  - *False Positives* (FP) – podaci negativne klase koji su pogrešno klasifikovani u pozitivnu
  - *False Negatives* (FN) – podaci pozitivne klase koji su pogrešno klasifikovani u negativnu

| | | Actual class | |
|---|---|---|---|
| | | **Cat** | **Non-cat** |
| **Predicted class** | **Cat** | 5 True Positives | 2 False Positives |
| | **Non-cat** | 3 False Negatives | 17 True Negatives |

# Ilustracija izgleda matrice zabune

Slika preuzeta sa: http://python-data-science.readthedocs.io/en/latest/evaluation.html

# Metrike za merenje performansi u binarnoj klasifikaciji

- Najčešće primenjivana metrika u slučaju rada sa podacima koji su izbalansirani po klasama jeste tačnost (engl. *accuracy*):

$$A = \frac{TP + TN}{TP + FP + TN + FN}$$

- Kada postoji mnogo više podataka jedne klase od podataka druge tačnost prestaje da bude korisna metrika
  - Praktično se ignorišu greške nad podacima malobrojnije klase dokle god se podaci klase čiji su podaci brojniji pravilno klasifikuju
    - Daje sjajnu ocenu performansi „modelu“ koji sve podatke klasifikuje tako da pripadaju klasi čiji su podaci brojniji
  - Ovo je sasvim suprotno od željenog ponašanja – malobrojnost podataka jedne klase obično znači da su greške nad tim podacima izuzetno važne
    - Primer – detekcija sarkazma u tekstu

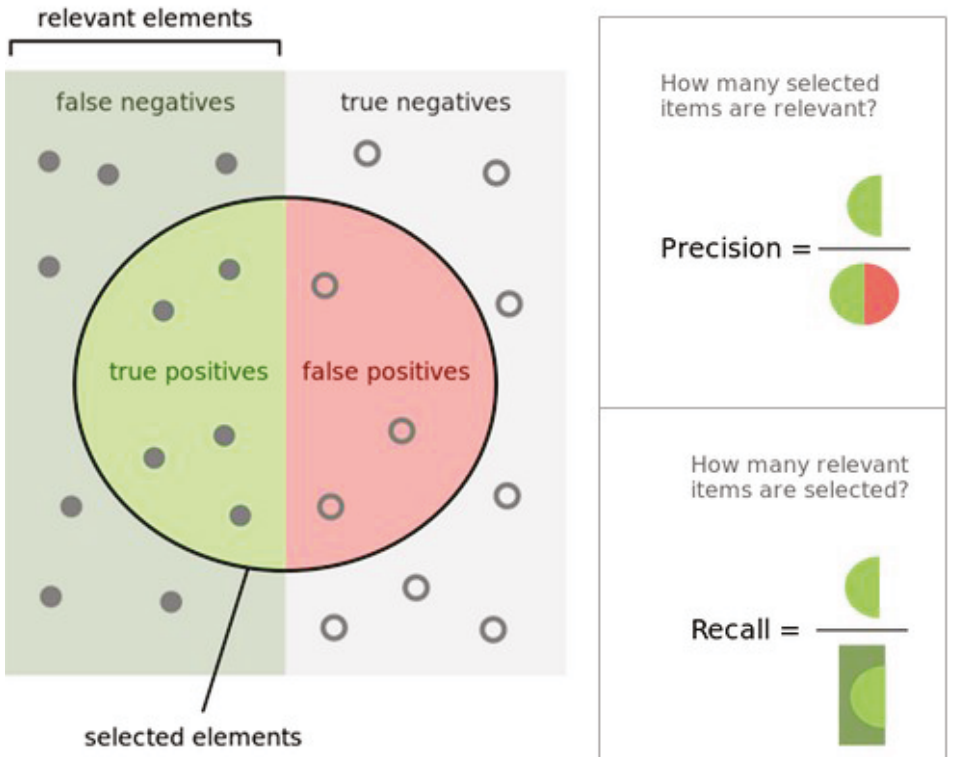
# Metrike za merenje performansi u binarnoj klasifikaciji

- Pri radu sa nebalansiranim podacima najčešće korišćena metrika je *F*-mera (engl. *F-measure*, negde se koristi i naziv *F1*-mera) koja predstavlja harmonijsku sredinu preciznosti (engl. *precision*) i odziva (engl. *recall*):

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$

$$F = \frac{2PR}{P + R}$$

- Kod preciznosti i odziva kao pozitivna klasa uzima se ona koja je malobrojnija, ako je skup neizbalansiran
- Ni *F*-mera, kao ni preciznost ni odziv uopšte ne uzimaju u obzir broj tačno klasifikovanih podataka negativne (tj. veće) klase – TN
- Lako je optimizovati preciznost na uštrb odziva i obrnuto




# Ilustracija preciznosti i odziva

Slika preuzeta sa: http://en.wikipedia.org/wiki/Precision_and_recall

# Primer izračunavanja metrika za merenje performansi u binarnoj klasifikaciji

$$A = \frac{5 + 17}{5 + 17 + 2 + 3} = 0.815$$

$$P = \frac{5}{5 + 2} = 0.714$$

$$R = \frac{5}{5 + 3} = 0.625$$

$$F = \frac{2 \times 0.714 \times 0.625}{0.714 + 0.625} = 0.666$$

| | | Actual class | |
|---|---|---|---|
| | | **Cat** | **Non-cat** |
| **Predicted class** | **Cat** | 5 True Positives | 2 False Positives |
| | **Non-cat** | 3 False Negatives | 17 True Negatives |

Slika preuzeta sa: http://python-data-science.readthedocs.io/en/latest/evaluation.html

# Metrike za merenje performansi u višeklasnoj klasifikaciji

- Tačnost (odnos broja pravilno klasifikovanih podataka i broja svih podataka) se može koristiti i u višeklasnoj klasifikaciji kada je skup podataka izbalansiran po klasama
- Često se koriste mikro-uprosečene i makro-uprosečene vrednosti preciznosti, odziva, i *F*-mere
- Mikro-uprosečavanje je uprosečavanje po podacima, gde je $k$ broj klasa:

$$P_{micro} = \frac{TP_1 + \cdots + TP_k}{TP_1 + \cdots + TP_k + FP_1 + \cdots + FP_k}$$

$$R_{micro} = \frac{TP_1 + \cdots + TP_k}{TP_1 + \cdots + TP_k + FN_1 + \cdots + FN_k} = P_{micro}$$

$$F_{micro} = \frac{2P_{micro}R_{micro}}{P_{micro} + R_{micro}} = P_{micro} = R_{micro} = A$$

# Metrike za merenje performansi u višeklasnoj klasifikaciji

- Makro-uprosečavanje je uprosečavanje binarnih vrednosti metrika po klasama:

$$P_{macro} = \frac{P_1 + \cdots P_k}{k} \qquad R_{macro} = \frac{R_1 + \cdots R_k}{k}$$

$$F_{macro} = \frac{2P_{macro}R_{macro}}{P_{macro} + R_{macro}}$$

- Pored ove definicije, na nekim mestima se $F_{macro}$ računa kao:

$$F_{macro} = \frac{F_1 + \cdots F_k}{k}$$

- Makro-uprosečavanje je pogodnije za nebalansirane skupove jer daje podjednaku težinu svim klasama

# Primer izračunavanja mikro-uprosečavanja

$$A = P_{micro} = R_{micro}$$

$$= \frac{5 + 3 + 11}{5 + 3 + 11 + 2 + 0 + 3 + 2 + 0 + 1}$$

$$= 0.7037$$

| | | Actual class | | |
|---|---|---|---|---|
| | | Cat | Dog | Rabbit |
| Predicted class | Cat | 5 | 2 | 0 |
| | Dog | 3 | 3 | 2 |
| | Rabbit | 0 | 1 | 11 |

Slika preuzeta sa: http://python-data-science.readthedocs.io/en/latest/evaluation.html

# Primer izračunavanja makro-uprosečavanja

$$P_{cat} = \frac{5}{5+2+0} = 0.714$$

$$R_{cat} = \frac{5}{5+3+0} = 0.625$$

$$P_{dog} = \frac{3}{3+3+2} = 0.375$$

$$R_{dog} = \frac{3}{2+3+1} = 0.5$$

$$P_{rabbit} = \frac{11}{0+1+11} = 0.917$$

$$R_{rabbit} = \frac{11}{0+2+11} = 0.846$$

| | | Actual class | | |
|---|---|---|---|---|
| | | Cat | Dog | Rabbit |
| Predicted class | Cat | 5 | 2 | 0 |
| | Dog | 3 | 3 | 2 |
| | Rabbit | 0 | 1 | 11 |

Slika preuzeta sa: http://python-data-science.readthedocs.io/en/latest/evaluation.html

# Primer izračunavanja makro-uprosečavanja

$$P_{macro} = \frac{P_{cat} + P_{dog} + P_{rabbit}}{3} = 0.669$$

$$R_{macro} = \frac{R_{cat} + R_{dog} + R_{rabbit}}{3} = 0.657$$

$$F_{macro} = \frac{2P_{macro}R_{macro}}{P_{macro} + R_{macro}} = 0.663$$

- Alternativno:

$$F_{macro} = \frac{F_{cat} + F_{dog} + F_{rabbit}}{3} = 0.658$$

| | | Actual class | | |
|---|---|---|---|---|
| | | Cat | Dog | Rabbit |
| Predicted class | Cat | 5 | 2 | 0 |
| | Dog | 3 | 3 | 2 |
| | Rabbit | 0 | 1 | 11 |

Slika preuzeta sa: http://python-data-science.readthedocs.io/en/latest/evaluation.html

# Metrike za merenje performansi u višeklasnoj klasifikaciji

- Kod drugog navedenog načina makro-uprosečavanja *F*-mere moguće je da njena vrednost ne bude jednaka harmonijskoj sredini preciznosti i odziva
- Pored mikro- i makro-uprosečavanja takođe se koristi i težinsko uprosečavanje, gde se metrike za svaku klasu ponderuju težinama određenim zastupljenošću te klase u celom skupu podataka
  - I u ovom pristupu *F*-mera može da ne bude jednaka harmonijskoj sredini preciznosti i odziva

# Procena performansi kreiranog modela

- Da bi se adekvatno procenilo koliko je kreirani model dobar, on se mora uporediti sa nekim osnovnim (engl. *baseline*) pristupom
- U praksi se koriste različiti osnovni pristupi
  - Klasifikacija
    - Odabir najfrekventnije klase – pristup koji svrstava svaki nov podatak u onu klasu koja se najčešće javlja u skupu za obučavanje
    - Odabir nasumične klase – pristup koji nasumično svrstava svaki nov podatak u neku od $k$ klasa
  - Regresija
    - Za svaki nov podatak se kao izlaz predviđa srednja vrednost izlaza svih podataka iz skupa za obučavanje

# Procena performansi kreiranog modela

- Navođenje samo rezultata modela bez ikakvog poređenja sa *baseline* pristupima nema smisla
  - Npr. tačnost od 90% na zadatku klasifikacije zvuči na prvi pogled jako dobro
  - Ali šta ako u posmatranom domenu 99% podataka pripada jednoj klasi?
    - Automatskim odabirom te klase dobija se tačnost od 99%
- Ispitivanje performansi osnovnih pristupa može pomoći i u zaključivanju da li je korišćena metrika za merenje performansi adekvatna ili ne