

Prediktiv analys

FÖRELÄSNING 10

Dagens fråga

- Vad är lagligt nu, men förmodligen inte om 25 år?

Dagens agenda

- Optimering av prediktionsmodeller
- Ensamble methods
 - Random forest
 - Bagging
 - Bossting och AdaBoost
- Natural Language Processing NLP

Förra föreläsning

- Data preprocessing
- Hantera NULL värden
- Outliers
- Feature scaling

- Python använder e för att skriva stora tal på kortare format
- e+06 betyder 10^6
- +06 är hur många nollor man ska lägga till
- Så $1.59\text{e}+07 = 15900000$

models	NULL	MLR	KNN	LASSO
train_mse	1.59239e+07	1.25084e+06	27.48	1.2509e+06
test_mse	1.58811e+07	1.2489e+06	583111	1.24941e+06

Root mean squared error (RMSE)

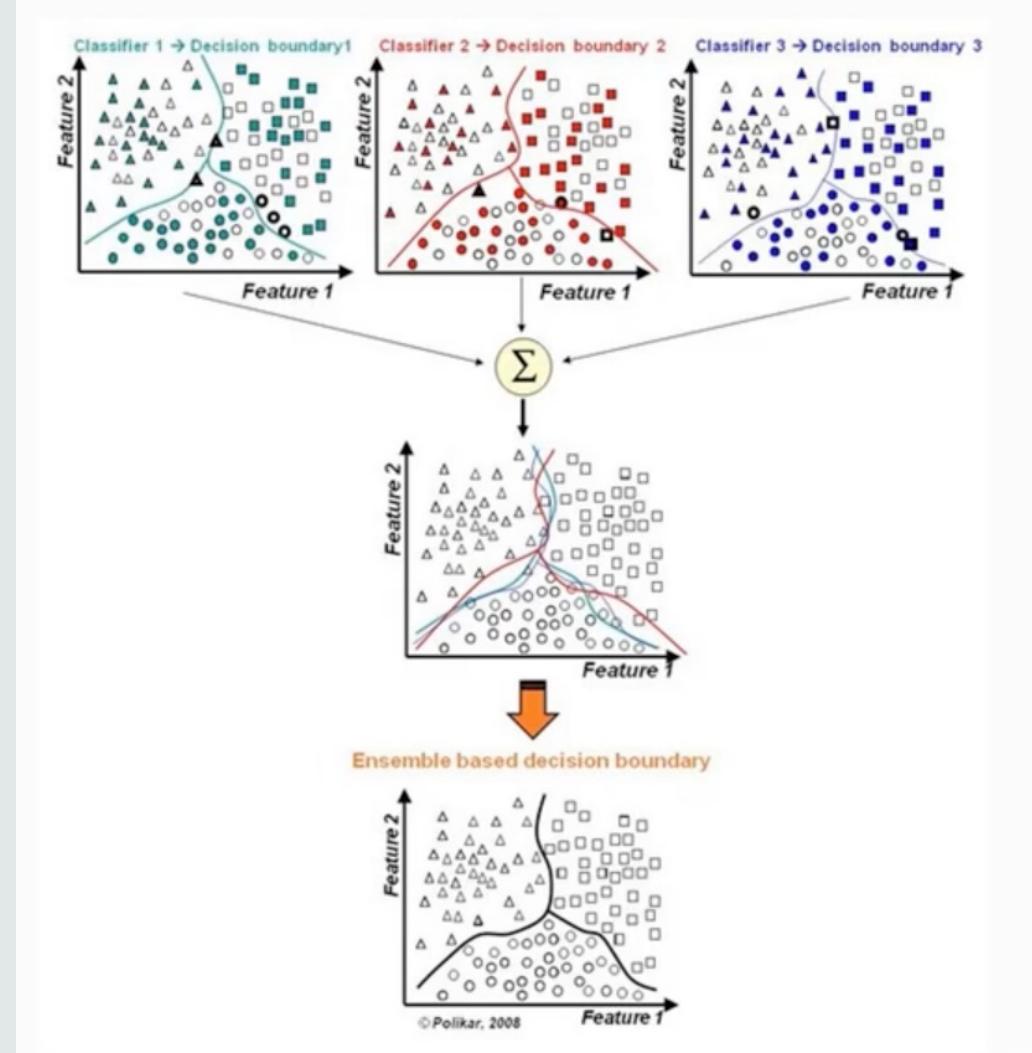
- Root mean squared error (RMSE)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{pred} - y_i)^2}$$

- Gör att error metrics är i samma skala som target y så vi lättare kan tolka resultatet.
- Python: `sklearn.metrics.mean_squared_error(y_true, y_pred, squared=False)`

Ensemble methods

- Ensemble methods – kombinera prediktioner från många individuella prediktioner för att öka prestanda.
- Bagging
- Random Forest
- Boosting & AdaBoost
- Komplexa modeller- tar längre tid att köra!

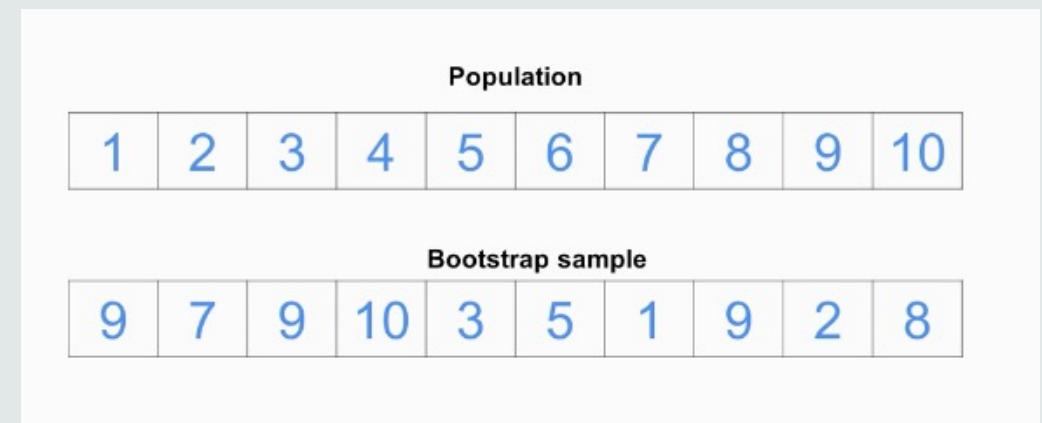


Resampling

- Vi önskar undersöka en population (som vi gör när ni använder statistik)
- Detta gör vi genom att hämta data om denna populationen med statistiska metoder, alltså **sampling**
- Tack vara samplet kan vi uppskatta hur populationen ser ut
- Resampling är tekniken att hämta ut flera mindre samples från just detta sample
- Med resampling kan vi med redan hämtat data förbättra uppskattningen om hur populationen ser ut
- Vi kan få statistik om samplet och större noggranhets

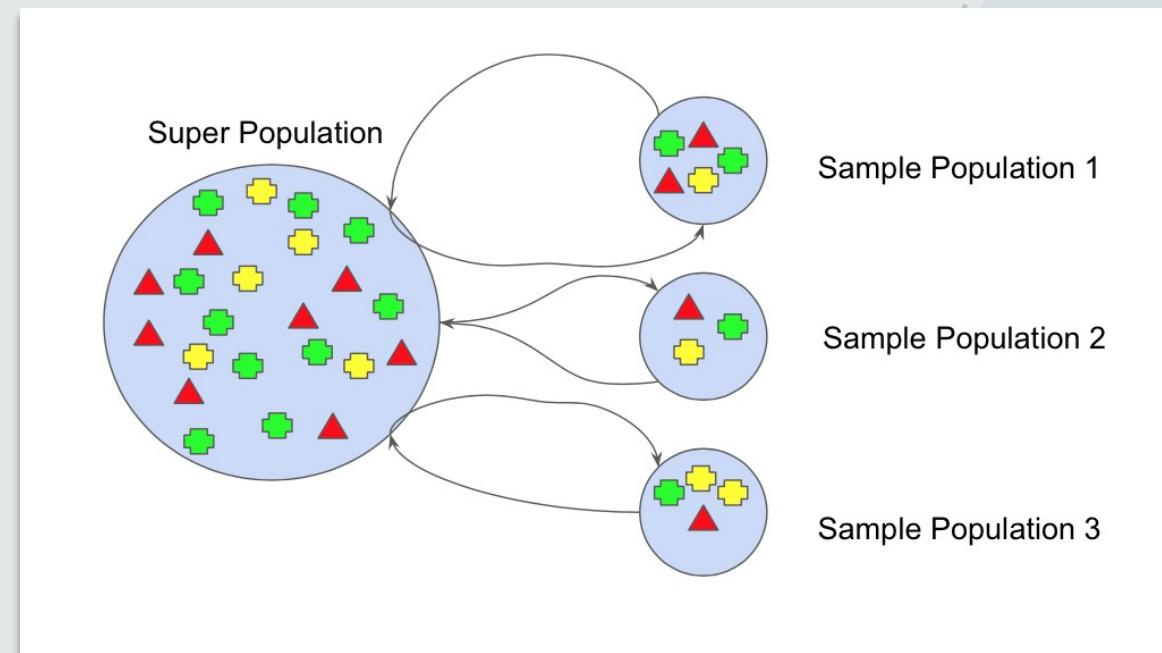
Bootstrapping

- Bootstrapping är en resample-teknik som används för att uppskatta statistik om en population.
- Bootstrapping – att sampla fram ett dataset med ersättning (sampling a dataset with replacement)
- Sampling with replacement är att vi random drar punkter från ett större dataset, en av gången, men returnerar punkten till datasetet. Därför kan samma observation dras flera gånger.



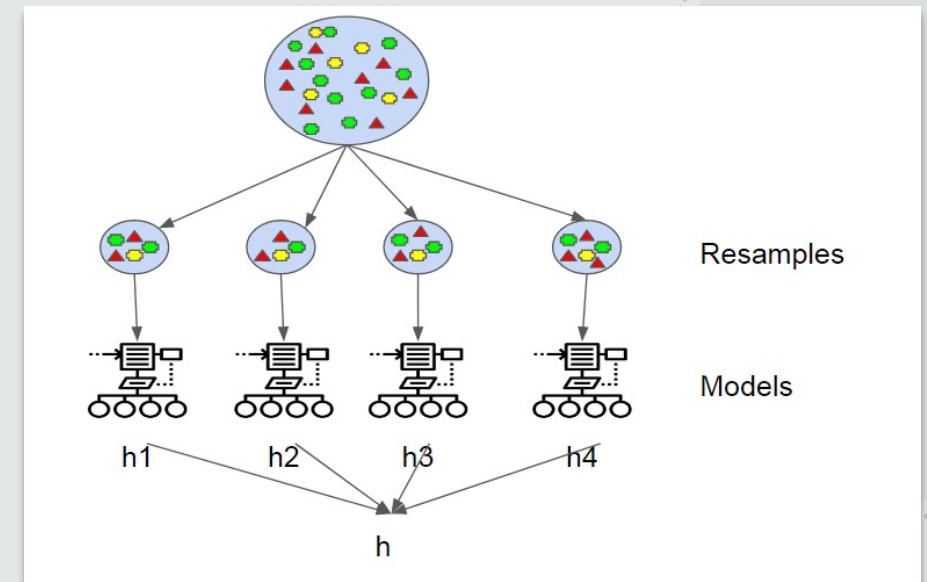
Bagging

- Bagging – också känt som **Bootstrap aggregation** (sammanslagning)
- Bagging är en metod för att **reducera variansen** i en Machine learning modell. Motverkar **overfitting**
- Baserad på bootstrapping – samplar ett antal dataset från träningsdatan
- Används ihop med regressions- och klassificeringsmodeller



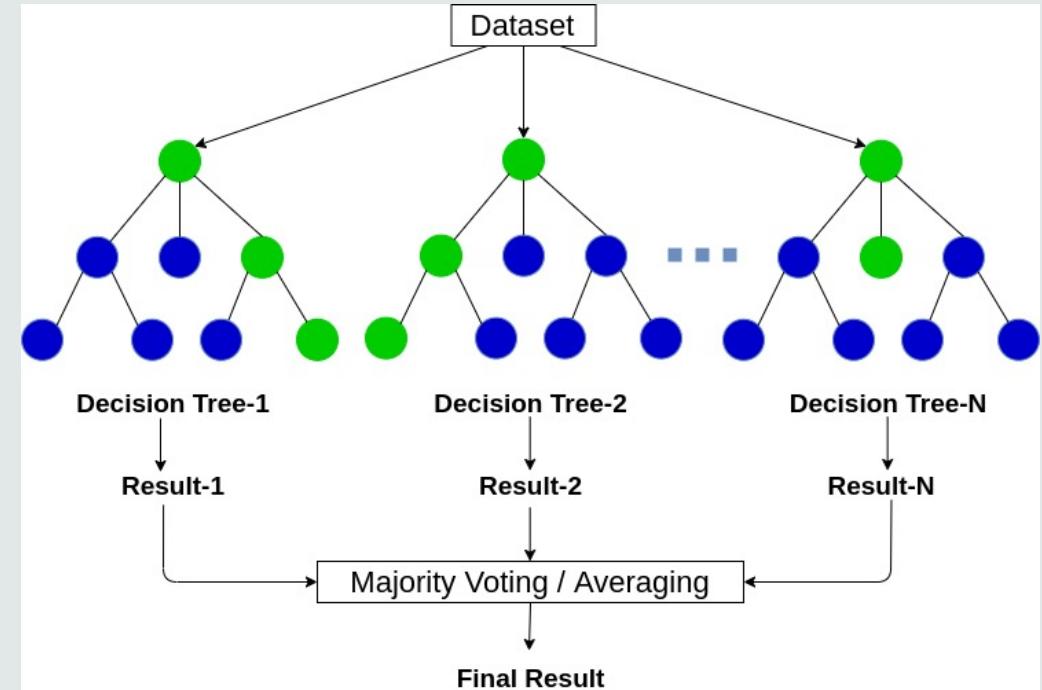
Bagging – steg

1. Välj antal modeller att använda
2. Sampla samma antal dataset från täningsdata
3. Fit varje modell till varje samplade dataset
4. Få *ensemble* prediktionen genom att *aggregera* all individuella prediktioner
 - Regression – använd genomsnittet av prediktionerna
 - Klassificering – majority vote (klassen som förekommer oftast)

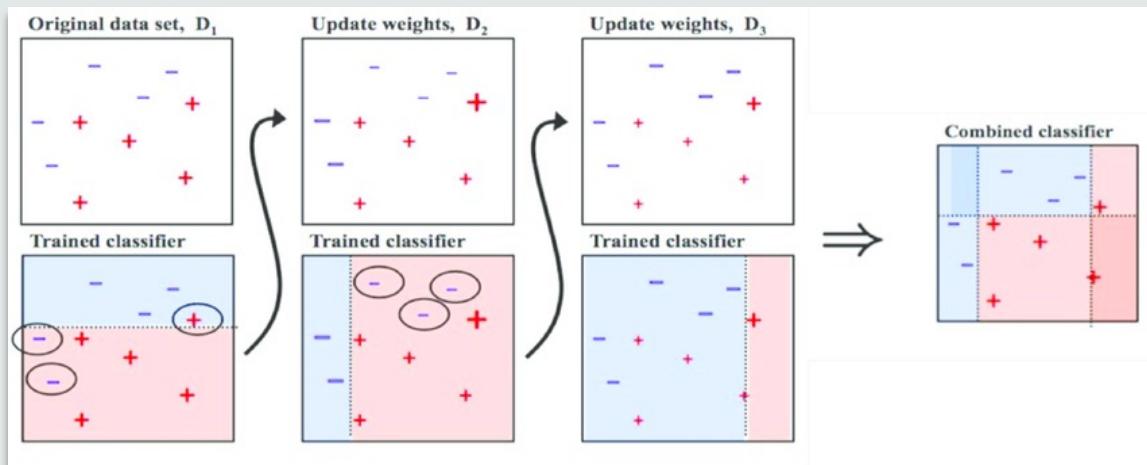


Radom Forest

- En ensemble metod för klassifierings/regressions-träd
- Likt bagging. Varje individuella träd tränas på ett bootstrapped sampel från träningsdatan
- Men när man splitter noder i trädet så är spliten som blir vald baserad på ett **random delmängd av featureserna**
- Detta gör varje prediktion något sämre (mer biased) men på grund av av decorrelation av prediktionerna så blir generellt random forest bättre.



Boosting & AdaBoost



- AdaBoost – Adaptive Boosting, Populär boosting
- Kan ett set svaga modeller skapa en stark modell?
- I stället för att träna många modeller individuellt gör vi det i en **sekvens**. Nästa modell är beroende på resultatet av föregående.
- AdaBoost re-weights all träningsdata i varje iteration baserad på föregående modells resultat.
- *Ex klassificering: fel klassificering får ökad vikt och korrekt klassificering får mindre vikt så att nästa modell fokuserar på de felklassificerade datapunkterna*

Regression med ensemble methods Python

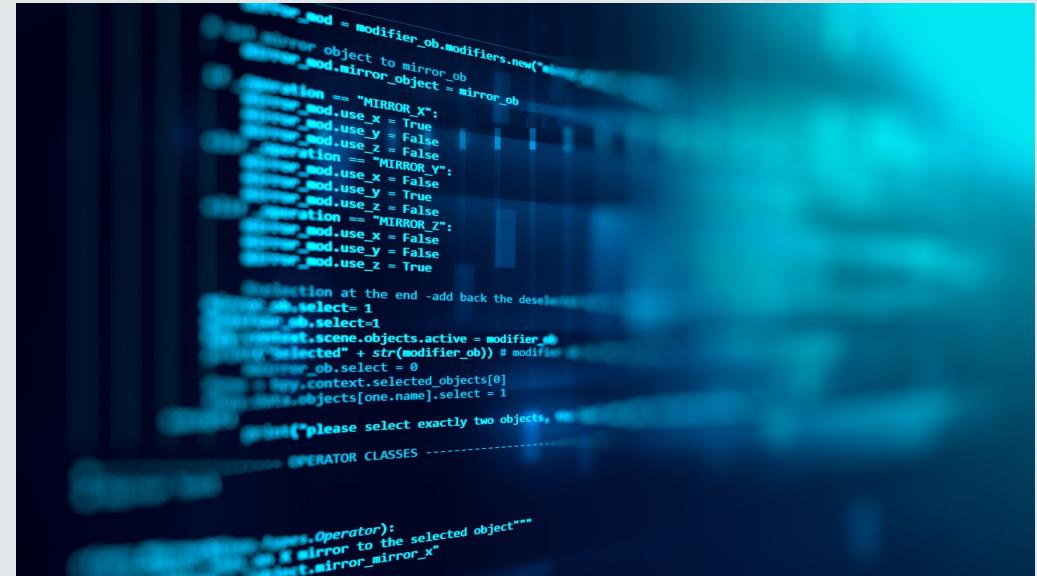
1. Diamant dataset
2. Träna en simpel KNN
3. Träna Bagging, Random Forest och Boosting
4. Jämföra resultaten

```
    mod = modifier_ob.modifiers.new("MIRROR")
    mod.mirror_object = mirror_ob
    if direction == "MIRROR_X":
        mod.use_x = True
        mod.use_y = False
        mod.use_z = False
    if direction == "MIRROR_Y":
        mod.use_x = False
        mod.use_y = True
        mod.use_z = False
    if direction == "MIRROR_Z":
        mod.use_x = False
        mod.use_y = False
        mod.use_z = True
    #selection at the end -add back the deselected
    select= 1
    select=1
    context.scene.objects.active = modifier_ob
    selected= str(modifier_ob)
    modifier_ob.select = 0
    bpy.context.selected_objects[0]
    bpy.context.selected_objects[one.name].select = 1
    print("please select exactly two objects, one to mirror and one to be mirrored")
    print("OPERATOR CLASSES -----")
    print("class Operator(bpy.types.Operator):
        bl_idname = "object.mirror_mirror_x"
        bl_label = "mirror to the selected object""")
```

EnsembleMethodsRegression.ipynb

Klassificering med ensemble methods Python

1. Credit card default dataset
 2. Träna en simpel Logistic regression modell
 3. Träna Bagging, Random Forest och Boosting
 4. Jämföra resultaten



EnsembleMethodsClassification.ipynb

”No free lunch”

Det är inte **en** modell
som alltid fungerar
bäst för varje problem
och varje dataset

Simple modeller kan
till exempel fungera
bättre än komplexa i
vissa fall ”No free
lunch” theorem



Natural language processing (NLP)

- NLP är den teknik som används för att få datorer att förstå människans språk
- Processen:
 1. En människa pratar med maskinen
 2. Maskinen fångar upp ljudet
 3. Konvertering av ljud till text
 4. Behandling av textens data
 5. Data till ljudkonvertering
 6. Maskinen svarar på människan genom att spela upp ljudfilen
- NPL ligger bakom kände applikationer som:
 1. Översättning som Google Translate
 2. Kontrollera grammatik som Microsoft Word och Grammarly
 3. Callcenter som Interactive Voice Response (IVR)
 4. Personlig assistens som OK Google, Siri, Cortana



Natural language processing (NLP)

- NPL är svårt då människans språk är komplex med regler svåra att fatta för maskinen. Datorn måste fatta båda orden och sammanhangen
- NLP identifierar och extraherar språkregler så att språkdata omvandlas till en form som datorer kan förstå. Algoritmen hämtar betydelsen till varje mening för att få information om den.
- Exempel på när det inte går: Från engelska "*The spirit is willing, but the flesh is weak.*" till russisk "*The vodka is good, but the meat is rotten.*"
- Tekniker:
 1. Syntax – ordningen av ord i en mening så de ger grammatisk mening. Dessa algoritmerna tillämpar grammatiska regler på ord för att hämta mening.
 2. Semantics – betydelsen av en text. Svårt att få till och ännu inte helt lösats. Algoritmerna för att förstå betydelsen och tolkningen av ord och hur meningar är strukturerade.

Spam Classifier

- Data set containing 5,572 Text Messages and their corresponding label (target):
- Spam or not spam

Hands on



Vad har vi gjort idag?

- Optimering av prediktionsmodeller
- Ensamble methods
 - Random forest
 - Bagging
 - Boosting och AdaBoost
- Natural Language Processing NLP

Nästa lektion

- Optimering av prediktionsmodeller
- Evaluering av modell: K-fold Cross Validation
- Hyperparameter tuning – hur välja bäst parameter till modellen?
- Exhaustive Grid Search
- Feature Selection