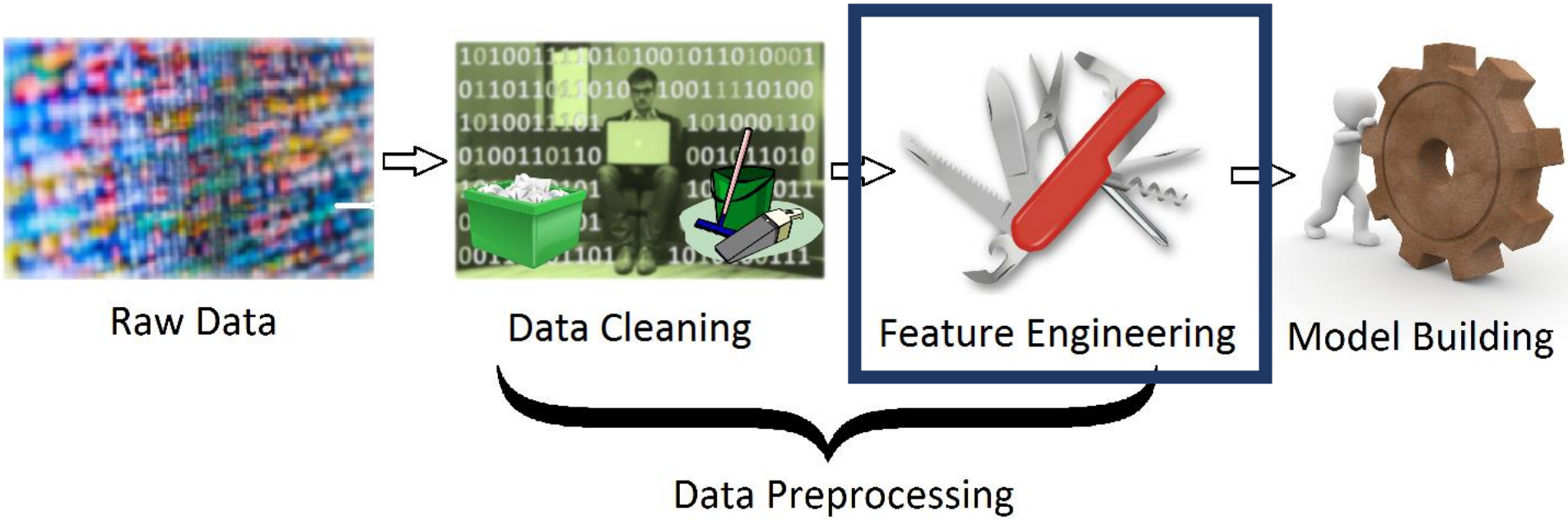
Raw Data
Data Cleaning
Feature Engineering
Model Building
Data Preprocessing

Obeležja

Informativnost obeležja

- Ukoliko atributi nisu dovoljno informativni, nijedan algoritam učenja ne može dati dobre rezultate
- Jedan način da ovo verifikujemo jeste da procenimo da li ljudski ekspert može da predvidi y ako mu je dato samo x
- Proveriti koje klase se međusobno mešaju i proveriti da li se može očekivati da postojeći atributi diskriminišu između njih
- Proveriti da li su atributi korelirani sa y pomoću koecijenta korelacije i grafika vrednosti y naspram x_d

Strategija

- Kada dizajniramo kompleksan sistem mašinskog učenja treba nam strategija – na čemu najpre da radimo kako bismo dobili najbolje moguće performanse

- Recimo da razvijamo sistem koji želi da kategoriše emailove u klase *spam* i *ham*

- Tekstualne podatke možemo predstaviti pomoću *bag-of-words* modela

Primer

- Logistička regresija postiže grešku od 2% na *spam* mailovoma i grešku od 2% na *ham* mailovima
- Ovo je neprihvatljivo velika greška za *ham* mailove
- SVM sa linearnim kernelom postiže 10% grešku na *spam* mailovima i grešku od 0.01% na *ham* mailovima i ove performanse su nam prihvatljive
- Sa druge strane, želimo da koristimo logistička regresiju jer je računarski efikasnija

Šta da radimo dalje?

- Da sakupimo puno podataka?
- Da formiramo obeležja bazirana na rutiranju dobijena iz *header*-a emaila?
- Da formiramo bolja obeležja ekstrahovana iz tela emaila?
  - Da li da “discount”/“discounts”, “deal”/”Dealer”, itd. tretiramo kao istu reč?
  - Da li da pravimo obeležja na osnovu interpunkcije?
- Da razvijemo sofisticirani algoritam za detekciju i korekciju grešaka u spelovanju?
  - Npr. “m0rtgage”, “med1cine”, “w4tches”

- Ovo možemo uraditi pomoću analize grešaka modela

Preporučen pristup

1. Započnite sa jednostavnim algoritmom koji možete brzo implementirati. Implementirajte ga i testirajte na validacionom skupu

2. Iscrtajte *learning curves* kako biste odlučili da li vam treba više podataka, više obeležja, sofisticiranija obeležja,...
   - Ovo je način da izbegnete „preuranjenu optimizaciju“ – u odluci na čega ćemo potrošiti vreme treba da nas vode dokazi, a ne samo „osećaj“

3. Analizirajte greške modela: testirajte model na validacionom skupu i (ručno) ispitajte greške koje vaš model pravi. Pokušajte da utvrdite da li postoji neki sistematičan trend u greškama
   - Ovo vas može inspirisati da konstruišete nova obeležja

Primer analize grešaka modela

- Recimo da imamo $N_{CV} = 500$ mailova u validacionom skupu i algoritam pogrešno klasifikuje 100 mailova

- Ručno kategorisati greške na osnovu **Tipa maila**
  - Npr. možemo videti da su loše klasifikovani spam mailovi vezani za prodaju lekova ili falsifikovanih satova ili se radi o *phishing* mailovima
  - Pharmacies: 12, Replica/fake: 4, Phishing: 53, Other: 31

  ➢ algoritam naročito loše radi na phishing mailovima, pogledati ove mailove i videti koja obeležja bi mogla pomoći

Tip maila

- Razmislite koji signali/obeležja bi mogli pomoći da algoritam korektno klasifikuje ove mailove

- Namerne greške u spelingu (m0rgage, med1cine,...): 5
- Neobične rute: 16
- Neobična interpunkcija (mnogo uzvičnika,...): 32

➢ U ovom slučaju greške u spelingu zvuče kao dovoljno redak fenomen na koji ne vredi trošiti vreme.

➢Sa druge strane, izgleda da dosta spammer-a koristi neobičnu interpunkciju pa vredi uložiti vreme da se formiraju sofisticiranija obeležja bazirana na interpunkciji

Primer analize grešaka modela

- Ono što želimo da identifikujemo su primeri koji su najteži za klasifikaciju
- Često će različiti obučavajući algoritmi naći da su iste kategorije primera teške za predikciju
- Zbog toga je preporučena brza implementacija jednostavnog algoritma uz analizu grešaka kako bismo identifikovali ovakve primere i pronašli način da ih adresiramo