Applied Data Analytics for Public Policy

Machine Learning Recap

Rayid Ghani University of Chicago @rayidghani

Types of Learning

Unsupervised

Supervised

Clustering
PCA
…

Classification
Regression
…

Supervised learning framework

$$y = f(x)$$

output

"Learned" function

Features/Predictors

future/generalization

Training or Learning: Find an f that minimizes error in recovering y

Slide credit: L. Lazebnik

How to solve a prediction problem

- Define and Create label (outcome variable)
- Define and Create Features (predictors)
- Create Training and Validation Sets
- Train model(s) on Training Set
- Validate model(s) on Validation Set

Python Recap

- Define and Create label (outcome variable)

1. Get people of interest:

   We want people who have exited already but are not in prison currently

   select docnbr from ildoc data (up to today) where admit date > *beginning of time* and latest exit date < *today*

Python Recap

- Define and Create label (outcome variable)

1. Get outcome labels for people of interest:
   We want 1 for people who get admitted again within 5 years and 0 otherwise

select case (when admit_date < (*today* + 5 years) then 1 else 0 end)
from docnbrs_of_interest left join ildoc.ildoc_admit_after_today

Docnbr (fake)	Label (5 years)
212344235	0
424324555	1
434299999	1

Python Recap

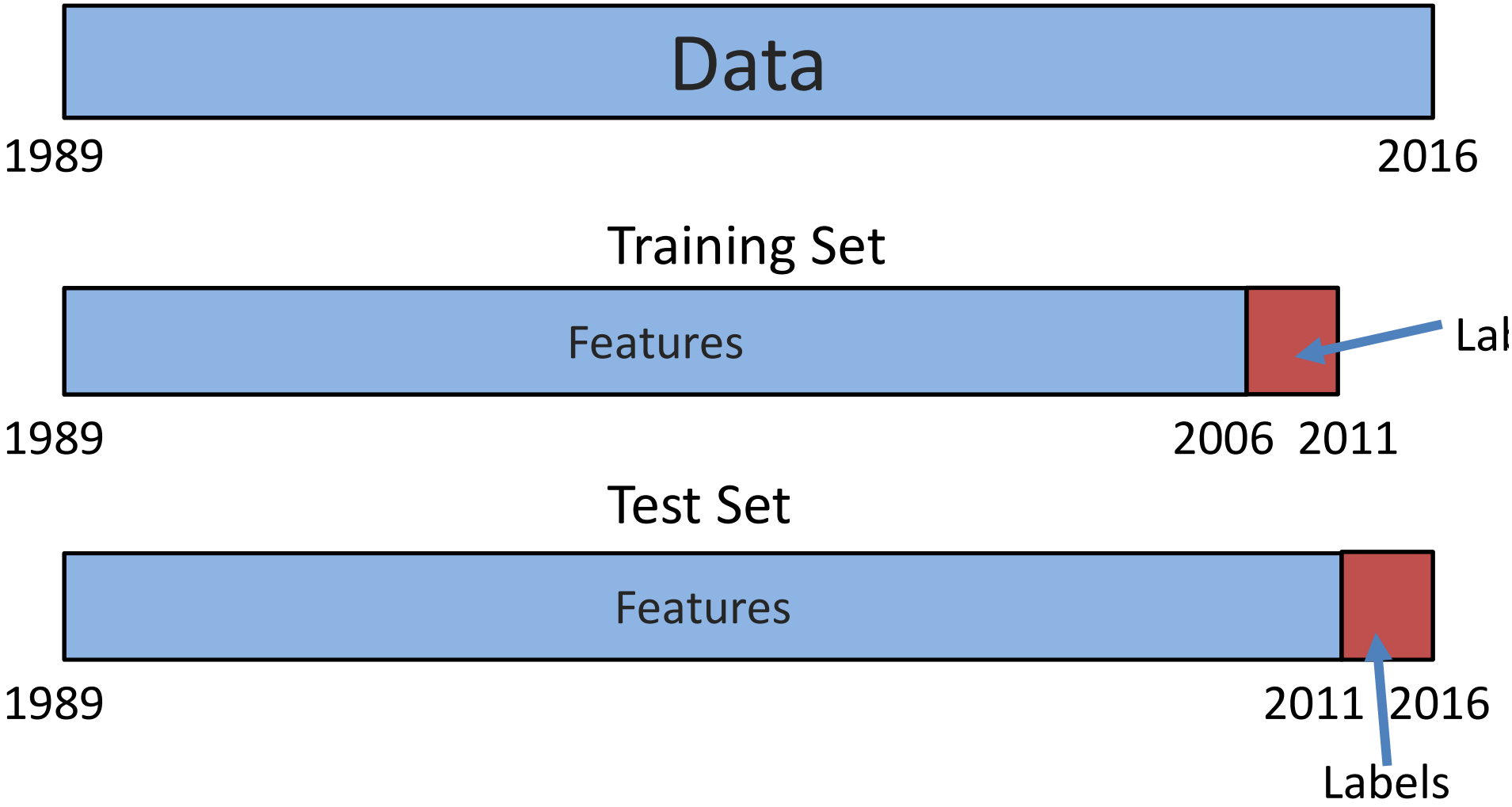
- Define and Create Features

1. Get people of interest
2. Get date (to create features as of)
3. Generate features

   select features from data where date < as_of_date

Docnbr (fake)	As_of_Date	Current_age	# of times admitted in the past 2 years
212344235	1/1/14	32	0
424324555	1/1/14	19	1
434299999	1/1/14	65	1

Create Training and Validation Sets



- Create labels and Features for Training and Test Sets

create_labels (2006, 5 years)

create_features (2006)

Train_matrix = join of the two above

create_labels (2011, 5 years)

create_features (2011)

test_matrix = join of the two above

- Train model(s) on Training Set

- Define feature columns X_train
- Define outcome column y_train
- Fit model

Model_type = LogisticRegression

Fitted_model = Model.fit(X_train, y_train)

- Test model(s) on Test Set

- Define feature columns X_test
- Define (known) outcome column y_test
- Score data using fitted model

Scores = Model.predict_proba(X_test)

Docnbr (fake)	scores
212344235	0.3
424324555	0.8
434299999	0.9

- Validate model(s) on Validation Set

- We have known outcomes (y_test) and predicted scores
- Turn scores into 0 or 1 using a threshold
- Calculate metrics
- Precision and recall at k graph

Calculate_precision_recall_at_k (scores, y_test, threshold)