DS2500
3/25-Tue.

Admin

3/28 - HW5 due 9pm
Lab 4-5 late deadline 9pm
Lecture 3:25pm / Zoom only

3/31 - review ex. practice
4/4 - Exam #2!
4/11 - second chance HW 9pm
4/15 - project deadline!

Agenda

1. multiple linear regression
2. evaluation of MLR
3. Python

1. multiple linear regression

↳ supervised learning

- something we know
- something to ask
— training data w/ known values

Simple Linear Regression

- need correlation first!
- X, Y values (training)
- predicts numeric values (targets)

KNN Classifier

- goal: classify based on features
- tweak by changing k
- euclidean for similarity

Logistic Regression

- binary classification
- probabilistic model
- slope varies

training / testing
X - independent var - features
Y - dep variable (target) - label

multiple linear regression

- ex- first linear regression
  X - adm rate (ind.)
  Y - tuition (dep) (target)

"real world" more than one X value influences y

simple: one x, one y

$y = mx + b$ (best fit)

multiple: $x_1, x_2, x_3, \ldots, x_n$, one y

$y = m_1 \cdot x_1 + m_2 \cdot x_2 + \ldots + m_n \cdot x_n + b$

$m_1, m_2 \ldots$ = regression coeffs

$x_1, x_2 \ldots$ = ind. variables

b = intercept

Python:

```
lr = LinearRegression()
lr.fit(X_train, y_train)
```

X_train → ind. variables; y_train → targets } training data

```
lr.predict(X_test)
```

↳ ind. variables (testing data)

Setting up data

↳ start w/ data all together

train_test_split to arbitrarily split

HW5/L205 ~ random_state = 0

today ~ no random state, randomize

train/test always same size,

values randomly chosen

y_test = actual targets

compare to predicted targets

2. Evaluation of MLR

↳ any supervised learning: compare actual vs. predicted

ex: KNN / Logistic ~ accuracy, precision, recall, F1

only have nominal data ~ were we wrong?

today: linear regression ~ all targets are numeric

how wrong were we? error

actual	predicted	error
1.5	2	.5
3.8	2.5	1.3

$MSE = \frac{(.5)^2 + (1.3)^2}{2}$

How can we improve our model to get a better MSE?

~ no k

↳ clean up the data

- remove null values
- correct data types
- normalization
- remove outliers
- remove a feature or combine
- weight features

$(.4)x + (.3)y + (.3)z$