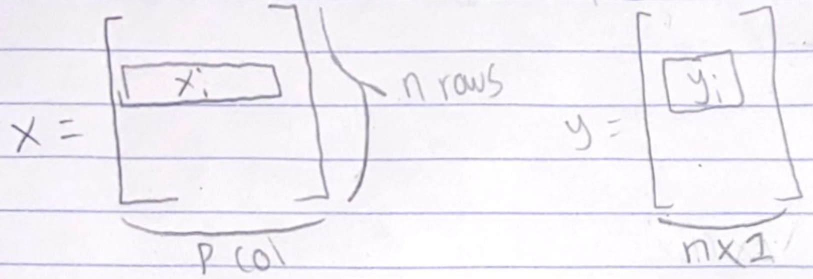
9/9 Lecture Notes

Admin: Change up seats, Lab 1 due tmr

Data repr. and featurization

Features (input) ex temp

Label (output) ex will you play outside



One value of y per data point, can be cont./discrete

Feature: name → shape
values → circle
vector → $x = [x_1, x_2, \cdots x_p]$

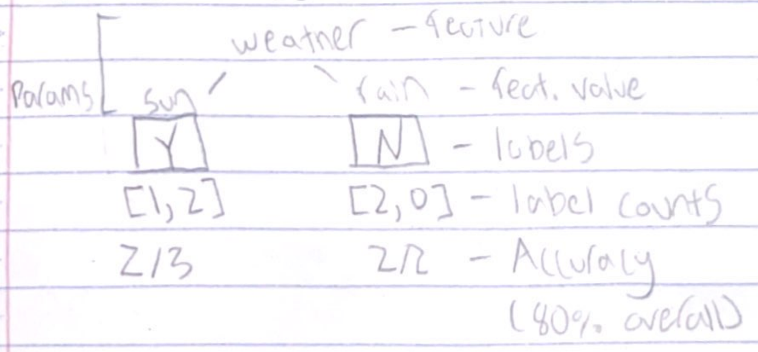
Featurization (make numerical) ex False → 0
↳ so comp. can interpret them True → 1

More ex. on slides Can map spectrum to values ex sunny → 1
rainy → 0

This process is done by Data Scientist to suit the model best.

Model: distribution that captures data

ex decision tree (on slide)

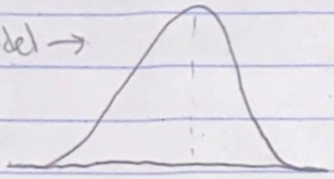


Data

weather	tennis
s	Y
r	N
r	N
s	Y
s	N

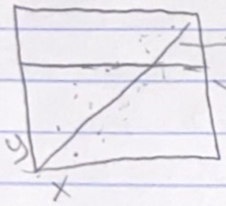
Normal distribution model

model →



mean] Model
Variance] Params.

Linear models:



→ Model (slope = m) $y = mx + b$

prediction based on model

m, b → model params

Handout thoughts: Q_2 - can featurize by binarizing (turning to 1s and 0s)
Q_4 - Model is "perfectly classified" depending on meaning
↳ if all given data points are classified
↳ if all possible data values are classified

Purpose of models:

Make predictions based on data

Vizualize data

Help humans make choices based on data

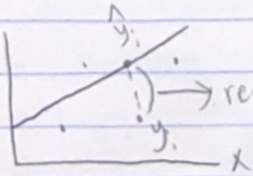
↳ ex, can see that a certain feature alligns more with output

Linear models:

feature $\vec{x}$ eq: $h_{\vec{w}}(x) = w_0 + w_1 x = \underset{\text{prediction}}{\hat{y}}$ residual: $y_i - \hat{y}_i$

output y

want to minimize RSS $\sum_{i=1} (y_i - \hat{y}_i)^2$
(error variance)



→ residual $(y_i - \hat{y}_i)$ → difference between prediction and data

***

Next lecture:

Use CL Args to input data

↳ parse_args package

Demo on website

Notes: can use – in arg name to make it optional

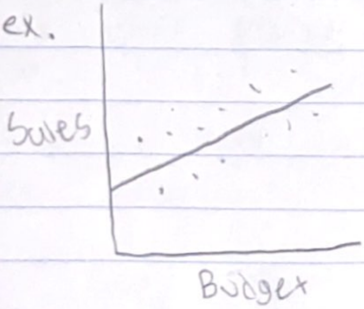
args are read in as string

Can set default arg value

Goals of fitting a linear model: 1) which features impact y

2) association between x and y

ex.



→ Model tells us as sales goes up, so does budget

Linear Regression

Used when output is continuous

Learned model: linear function mapping x to y (including $\underbrace{\text{feature weights}}_{\text{model params}}$ and bias)

Goal: to minimize RSS/SSE (error)

"simple" lin. regress. → one feature

$h_{\vec{w}} = w_0 + w_1 x = \hat{y}$

weights

$\vec{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$

Linear model may not always be most accurate
↳ could use polynomial curve to better fit data
MSE (average error)

Performance may vary from training data to testing data
"Overfitting" - over performs on training data, underperforms on new data