Teaching Elements of Machine Learning in A Quantitative Reasoning Course

Mutiara Sondjaja

New York University

Plug and Play Data Science Lessons
MathFest 2019

Slides and lesson materials can be found on: cims.nyu.edu/∼sondjaja/teaching and github.com/tiasondjaja/plugplaydslessons

A (Movie) Classification Project

The culmination of a 2-3 week module in a semester-long data- and modeling-centric quantitative reasoning course, taught in Fall 2018 at NYU.

A (Movie) Classification Project

The culmination of a 2-3 week module in a semester-long data- and modeling-centric quantitative reasoning course, taught in Fall 2018 at NYU.

Wanted: A project that

- invites students to get their hands dirty with data
- is accessible: high-level ideas are intuitive
- allows free exploration
- has a high ceiling: room for technical and creative growth; removable scaffolding for more advanced students
- highlights course theme: Data → Model → Decision/Prediction.

Tools

1. R/Python
   - R (with dplyr and ggplot2) or
     python (with pandas and matplotlib)
   - Emphases on data-centric exploration, general programming elements, and quantitative thinking as opposed to memorizing syntax
   - creating, interpreting, working with data visualizations
   - understanding how to work with variables and functions; conditional statements and boolean expressions; loops

2. Cloud-based Jupyter Notebooks

   Options:

   - Google Collaboratory
   - CoCalc.com (formerly Sage Math Cloud)
   - Other JupyterHubs or set up your own JupyterHub (e.g., https://tljh.jupyter.org), etc.

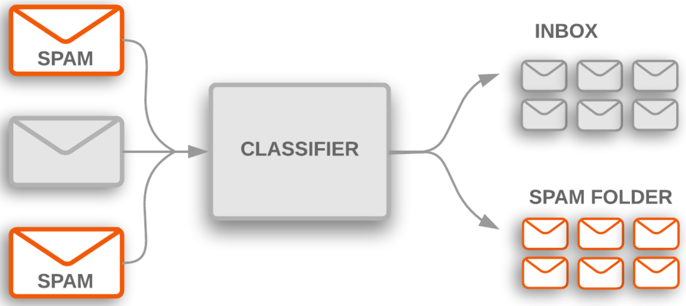
Classification

Task: Classify data point into one of several categories.

Classification

Task: Classify data point into one of several categories.

Examples:

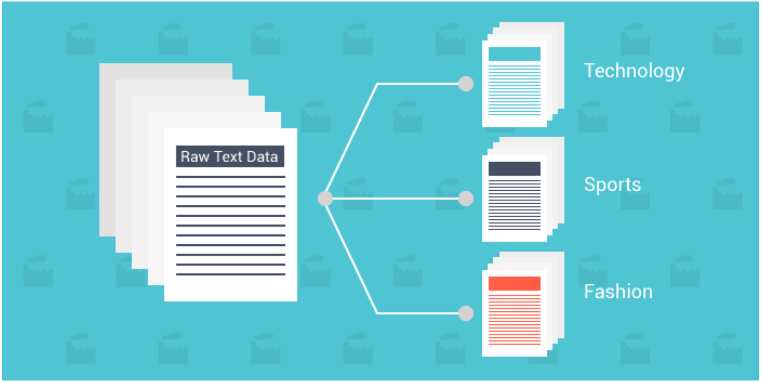


https://developers.google.com/machine-learning/guides/text-classification/

Classification

Task: Classify data point into one of several categories.

Examples:

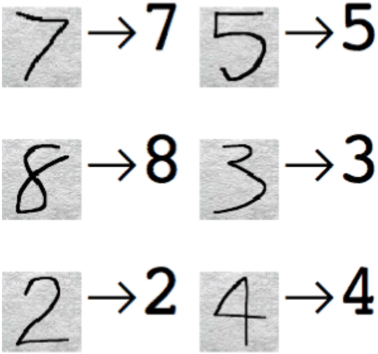


https://miro.medium.com/max/960/1*HgXA9v1EsqlrRDaC_iORhQ.png

Classification

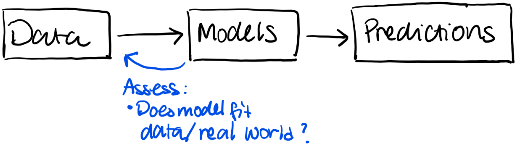
Task: Classify data point into one of several categories.

Examples:

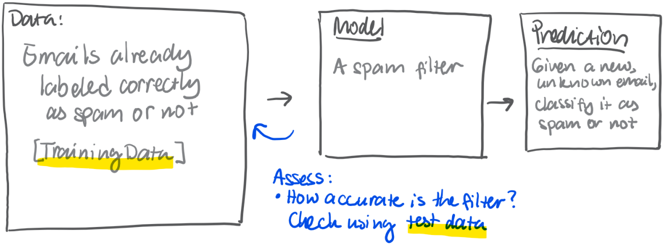


https://www.wolfram.com/mathematica/new-in-10/enhanced-image-processing/HTMLImages.en/handwritten-digits-classification/smallthumb_10.gif

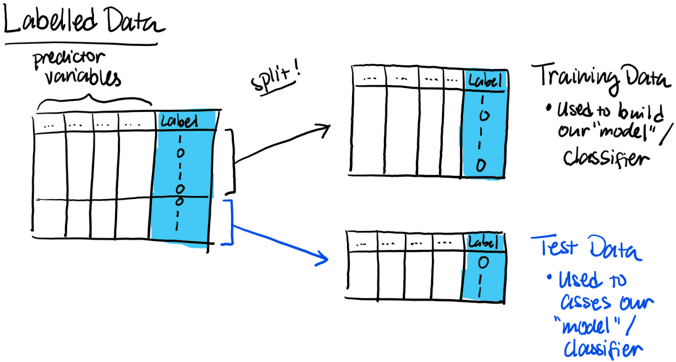
Classification



Example


Data:
Emails already labeled correctly as spam or not
[Training Data]
Model
A spam filter
Prediction
Given a new, unknown email, classify it as spam or not
Assess:
• How accurate is the filter? Check using test data

Classification



Training Data

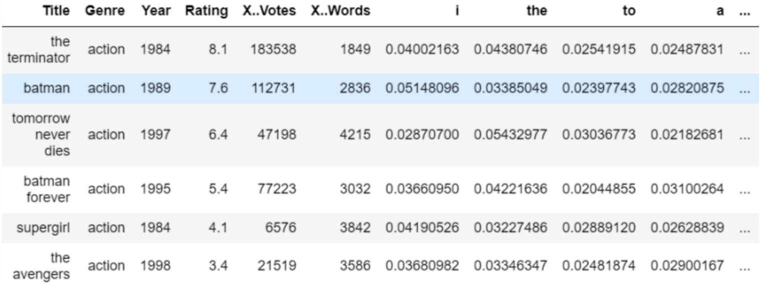
- Used to build our "model"/ classifier

Test Data

- Used to asses our "model"/ classifier

The Project

Predict a movie's genre (romance vs. action) based on the frequencies of words.



Title	Genre	Year	Rating	X..Votes	X..Words	i	the	to	a	...
the terminator	action	1984	8.1	183538	1849	0.04002163	0.04380746	0.02541915	0.02487831	...
batman	action	1989	7.6	112731	2836	0.05148096	0.03385049	0.02397743	0.02820875	...
tomorrow never dies	action	1997	6.4	47198	4215	0.02870700	0.05432977	0.03036773	0.02182681	...
batman forever	action	1995	5.4	77223	3032	0.03660950	0.04221636	0.02044855	0.03100264	...
supergirl	action	1984	4.1	6576	3842	0.04190526	0.03227486	0.02889120	0.02628839	...
the avengers	action	1998	3.4	21519	3586	0.03680982	0.03346347	0.02481874	0.02900167	...

242 rows, 5006 columns
(242 movies; 6 info cols + 5000 word frequencies)

The Project

Predict a movie's genre (romance vs. action) based on the frequencies of words.

Project Components

0. Split data into training and test data
1. "Sniff around"
   understand the dataset; plot some visualizations; find patterns in data
2. Use data to build models
   a. "Simple Classifiers"
   b. k-Nearest Neighbor Classifiers
3. Assess the models
4. Lightning talk presentations and written report

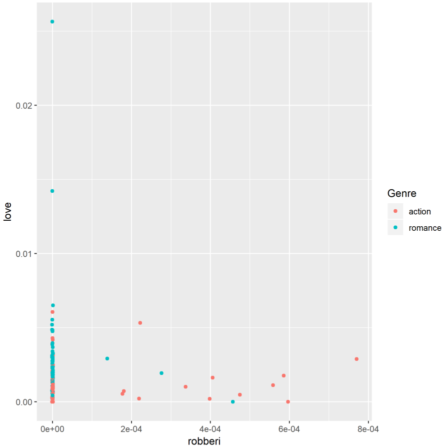
The Project

Predict a movie's genre (romance vs. action) based on the frequencies of words.

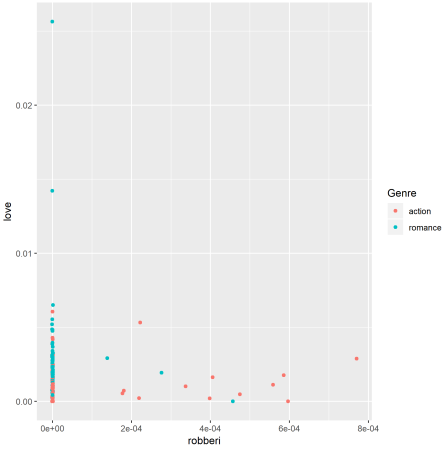
Project Components

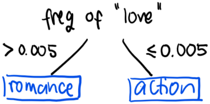
0. Split data into training and test data
1. "Sniff around"
   understand the dataset; plot some visualizations; find patterns in data
2. Use data to build models
   a. "Simple Classifiers"
   b. k-Nearest Neighbor Classifiers
3. Assess the models
4. Lightning talk presentations and written report

Exploring Data and Building “Simple Classifiers”



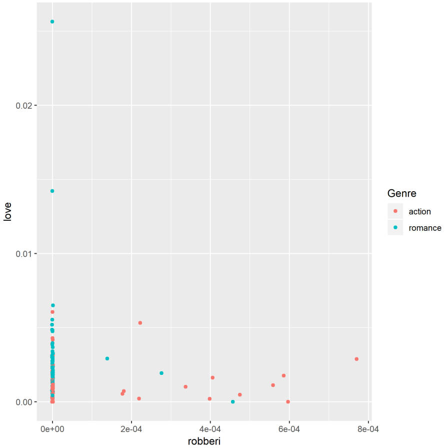
Exploring Data and Building "Simple Classifiers"

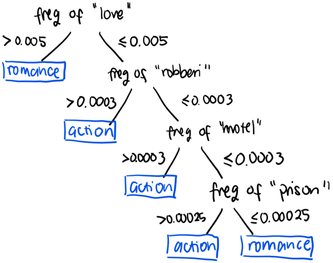




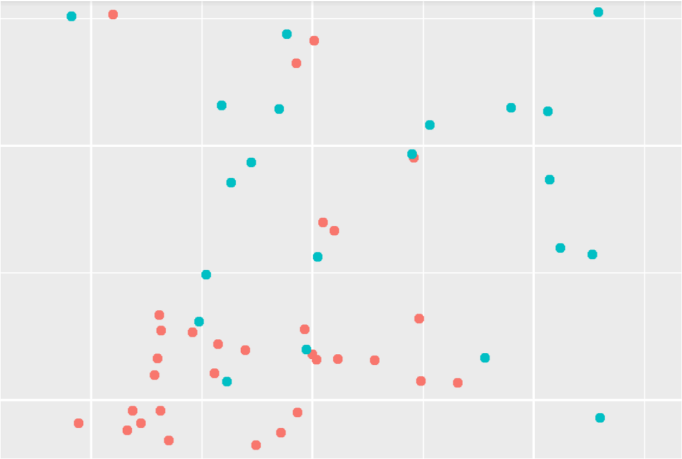
```
if (freq_love > 0.005)
        genre ← 'romance'
else
        genre ← 'action'
```

Exploring Data and Building “Simple Classifiers”

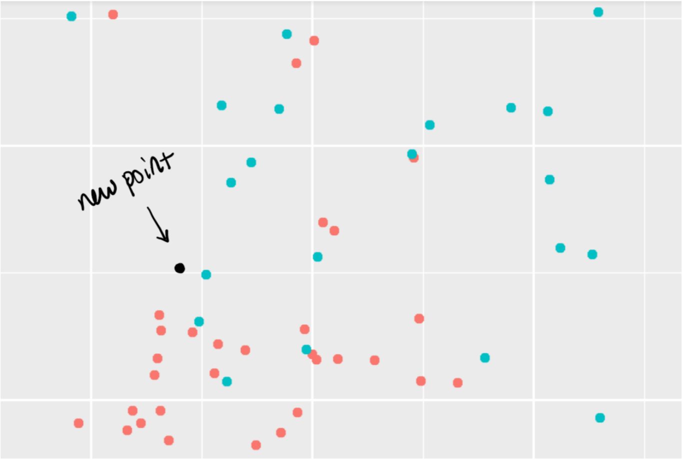




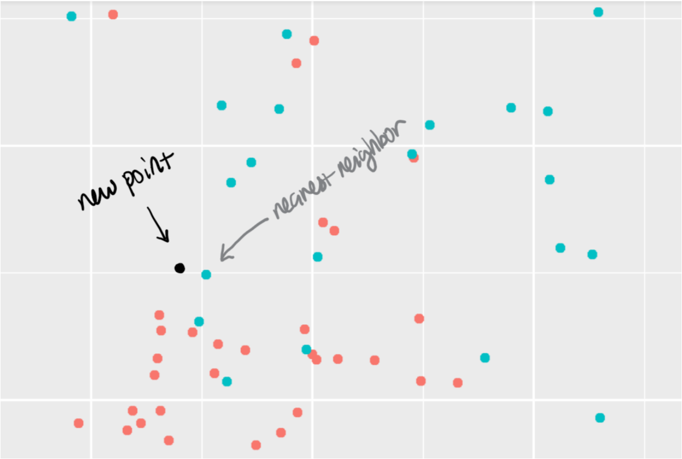
k-Nearest Neighbors



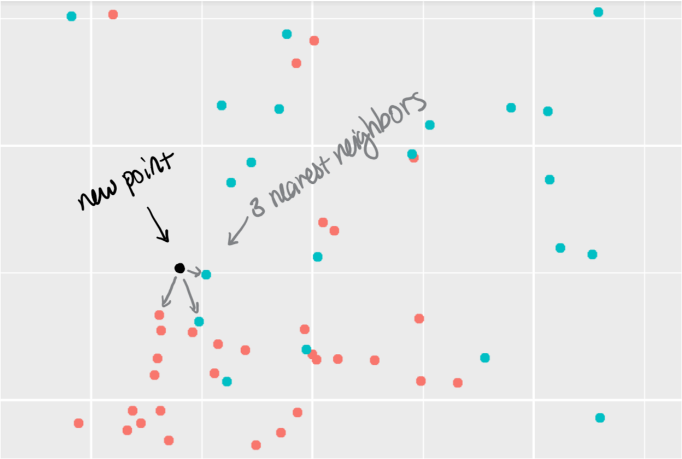
k-Nearest Neighbors



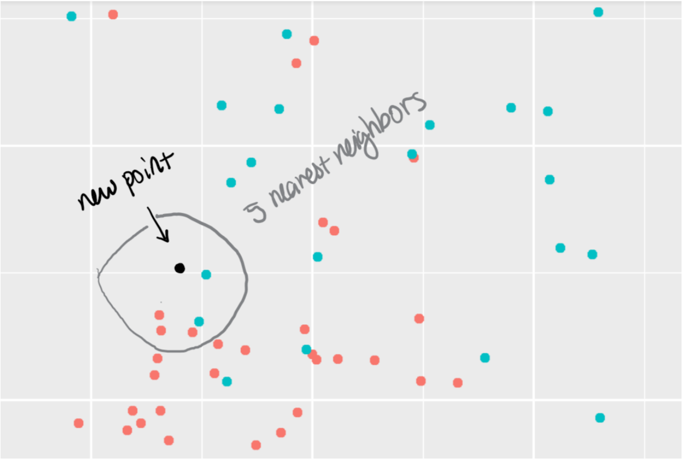
k-Nearest Neighbors



k-Nearest Neighbors



k-Nearest Neighbors

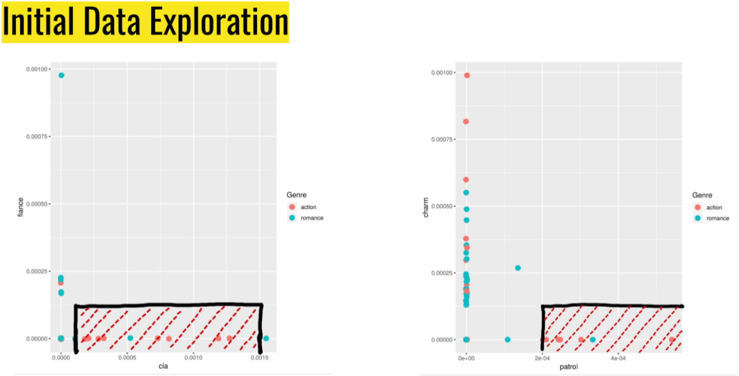


Demonstration

Sample completed project
Notebook

Lightning talk progress reports
Student presentation 1
Student presentation 2

Student Presentation 1

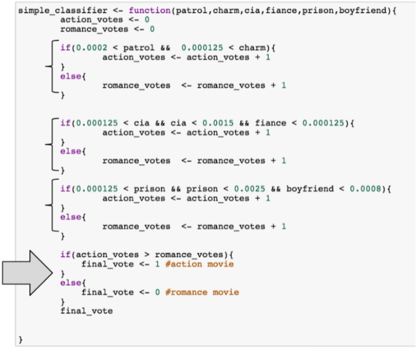


Initial Data Exploration

Student Presentation 1

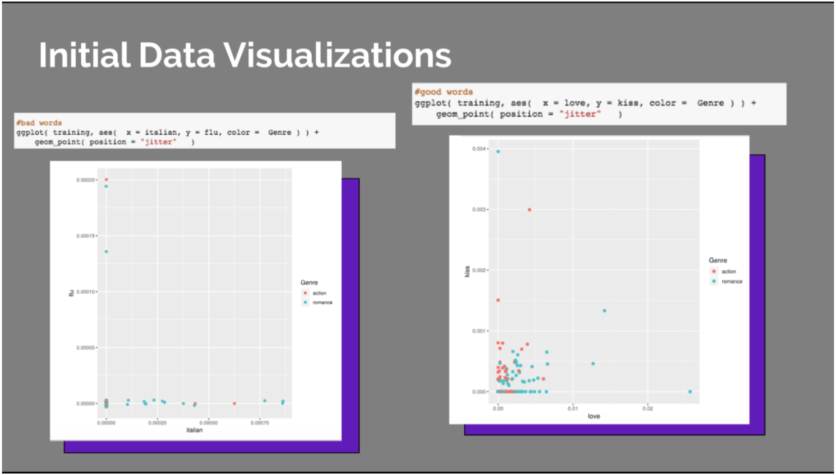
Simple Classifier

- Majority rules model
- 3 pairs of if-else statements
- Tally up votes



```
simple_classifier <- function(patrol,charm,cia,fiance,prison,boyfriend){
        action_votes <- 0
        romance_votes <- 0

        if(0.0002 < patrol &&  0.000125 < charm){
                action_votes <- action_votes + 1
        }
        else{
                romance_votes  <- romance_votes + 1
        }


        if(0.000125 < cia && cia < 0.0015 && fiance < 0.000125){
                action_votes <- action_votes + 1
        }
        else{
                romance_votes  <- romance_votes + 1
        }
        if(0.000125 < prison && prison < 0.0025 && boyfriend < 0.0008){
                action_votes <- action_votes + 1
        }
        else{
                romance_votes  <- romance_votes + 1
        }

        if(action_votes > romance_votes){
            final_vote <- 1 #action movie
        }
        else{
            final_vote <- 0 #romance movie
        }
        final_vote


}
```

Student Presentation 2



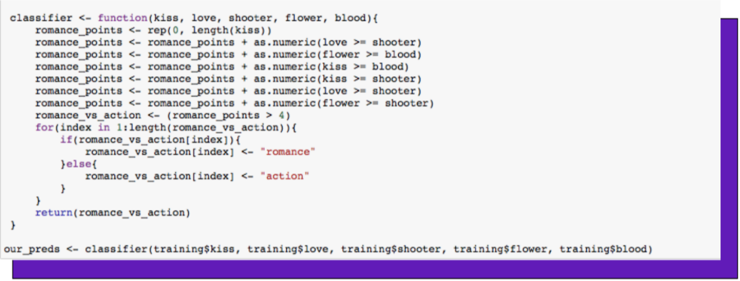
Initial Data Visualizations

```
#bad words
ggplot( training, aes(  x = italian, y = flu, color =  Genre ) ) +
    geom_point( position = "jitter"   )
```

```
#good words
ggplot( training, aes(  x = love, y = kiss, color =  Genre ) ) +
    geom_point( position = "jitter"    )
```

Student Presentation 2

Outline of Simple Classifier:



```
classifier <- function(kiss, love, shooter, flower, blood){
    romance_points <- rep(0, length(kiss))
    romance_points <- romance_points + as.numeric(love >= shooter)
    romance_points <- romance_points + as.numeric(flower >= blood)
    romance_points <- romance_points + as.numeric(kiss >= blood)
    romance_points <- romance_points + as.numeric(kiss >= shooter)
    romance_points <- romance_points + as.numeric(love >= shooter)
    romance_points <- romance_points + as.numeric(flower >= shooter)
    romance_vs_action <- (romance_points > 4)
    for(index in 1:length(romance_vs_action)){
        if(romance_vs_action[index]){
            romance_vs_action[index] <- "romance"
        }else{
            romance_vs_action[index] <- "action"
        }
    }
    return(romance_vs_action)
}

our_preds <- classifier(training$kiss, training$love, training$shooter, training$flower, training$blood)
```

This project

- invites students to get their hands dirty with data
- is accessible: high-level ideas are intuitive
- allows free exploration
- has a high ceiling: room for technical and creative growth; removable scaffolding for more advanced students
- highlights course theme: Data → Model → Decision/Prediction.

Thank you!

Slides and lesson materials:
`cims.nyu.edu/~sondjaja/teaching`
`github.com/tiasondjaja/plugplaydslessons`

Email:
sondjaja@nyu.edu

Acknowledgments

This project was inspired by a course project in UC Berkeley's "Data Science for Everyone" course.
Among others, we modified the project, course emphases, and goals to that of a QR course, infused a greater degree of free exploration (afforded by smaller class sizes), and adapted material from python to R.