Making ends meet: Projecting demand for class seats at a major American university

Nyssa Bulkes
Springboard Data Science Career Track
Capstone #2 Presentation Slides

Introduction

- One of the biggest challenges facing university administrators and department heads: Meeting student needs
- As a business, university success = student success
  - Ease, availability of required classes
  - Graduation rates
- Course, seat availability is part of the university's product students pay for
  - If the product is sparse or difficult to obtain, students will go elsewhere

Introduction

- Providing seats in balanced, economical way requires calculation
  - Need to balance student-to-instructor ratio with not “over-offering”

- Seat projection requires attention to detail
  - Entry-level or advanced course? Which semester?
  - Specificity facilitates both accuracy and user experience

- **In the current project**, I will:
  - Demonstrate how different factors—i.e. semester type—modulate enrollment patterns expected for a given course
  - Use data-science techniques to show how these patterns can be computationally modelled

The scope

- University of Arizona
  - One of three major public universities in Arizona
  - Current undergraduate population: 35,123
- Seat projections as a service
  - Dashboard with prompts to specify course parameters needed for projection
- Project goals:
  - Demonstrate how course-specific factors can modulate enrollment patterns
  - Exemplify how such a projection tool could be valuable to those responsible for course scheduling, planning

The data

- 30 years of enrollment data for ENGL101
  - 100-level, general-education course required for bachelor's degree of any kind
  - Taken early in undergraduate career to facilitate sign-up in more advanced courses
  - Volume of students must be accounted for
- Data is deidentified
  - Aggregated at the level of class enrollment; cannot be traced back to any particular student
- Fall-, summer-, and spring-semester data
  - ENGL101 not offered in winter during 30-year period
- Cleaning largely unnecessary
  - Data pulled from university database by me, as an analyst at UAIR
  - No rows with null values

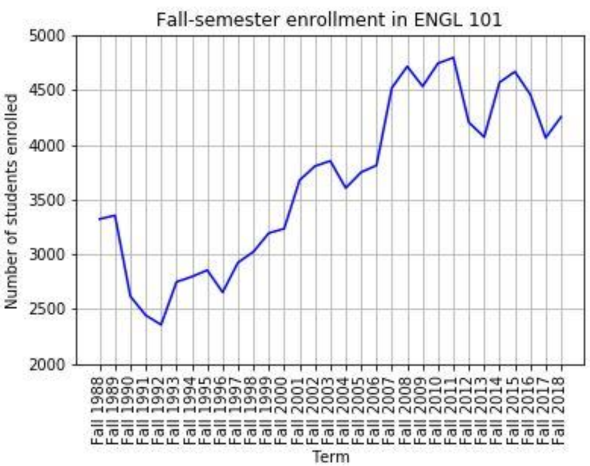
The application

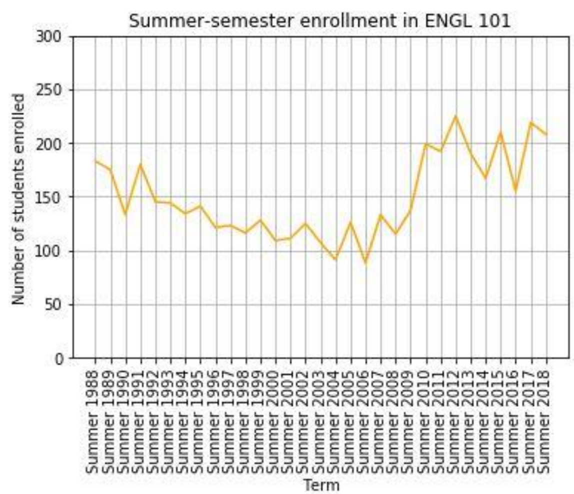
- Dashboard as user interface
  - Course facilitators able to enter course-specific parameters in prompts (i.e. semester type, specific course catalog number)

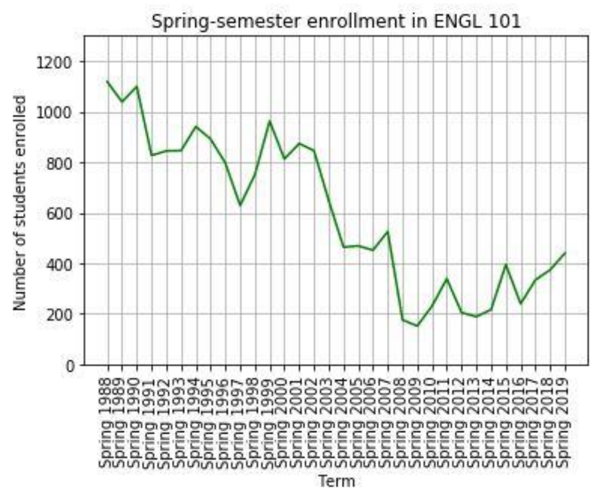
- By-semester projection capability necessitates subsetting data by semester
  - Improved model accuracy, reduced noise
  - In future, extend tool to other courses, i.e. different course types

Exploratory data analysis

- Data subsetted, read into pandas for wrangling
- Visualizations for fall, summer, spring ENGL101 data







Model selection

- Univariate time-series
  - OLS inappropriate: Enrollment is count data, cannot be fraction or negative value
- Step 1: AR, auto-regressive, model

$$X_t = \delta + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-1} + A_t$$

- Assumptions:
  - Stationarity: Calculate means, variances to determine equality
    - Not met; Data need to be differenced until stationarity is achieved
  - Normality: Shapiro-Wilks test
    - Met
  - Trend: Autocorrelation plots
    - Points related but not strong correlation

Model training

- AR model: 70/30 train-test split
  - Enormous MSE, poor accuracy suggest model is severely underfit

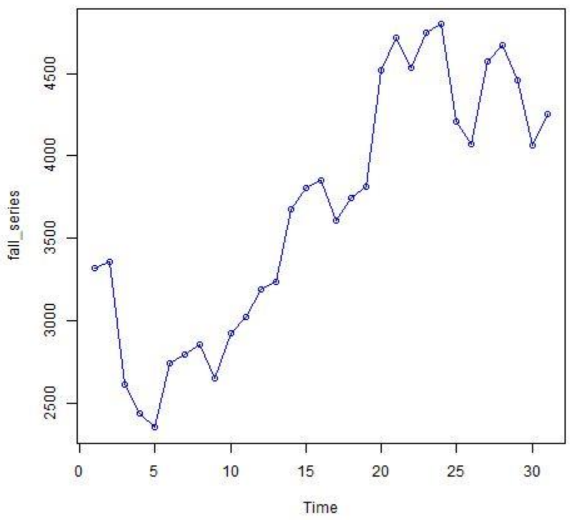
- ARIMA (auto-regressive, integrated, moving-average) may be a better fit for data we know not to be random (AR models typically used to model random processes)
  - ARIMA accommodates non-stationarity via differencing

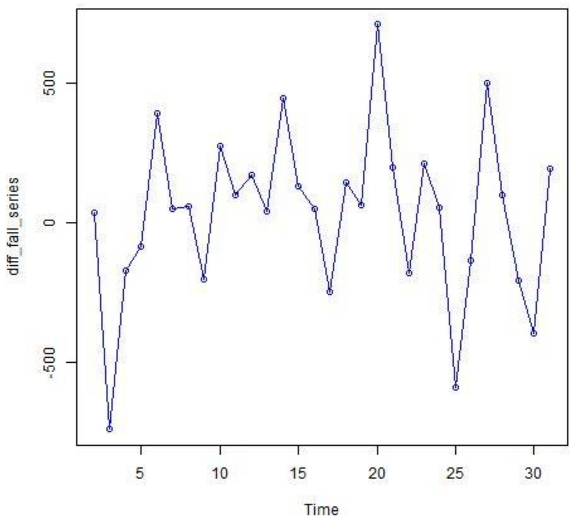
$$y_t = \delta + \{\phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p}\} + \{\theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}\} + \epsilon_t$$

$$\implies y_t = \delta + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t$$

ARIMA, by hand

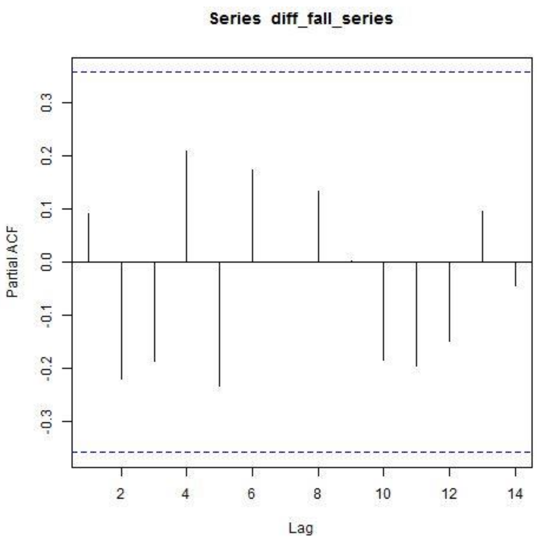
- “I” parameter *(D)* = number of differences needed to achieve stationarity
  - *1 differencing for fall data* → *d = 1*

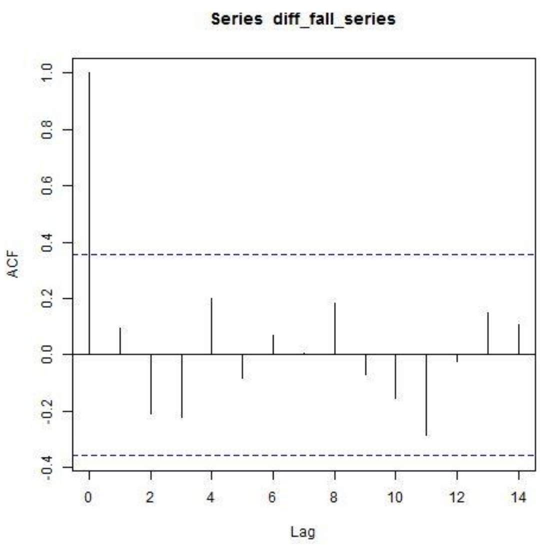




ARIMA, by hand (cont.)

- "AR" parameter *(P)* = PACF
  - Drop-off after 1 lag; $p = 1$
- "MA" parameter *(Q)* = ACF
  - Drop-off after 1 lag; $q = 1$





Model evaluation

- ARIMA, by-hand*:
  - Best fall model: ARIMA(1,1,0)
  - Best summer model: ARIMA(0,1,1)
  - Best spring model: ARIMA(0,1,0)

  *AIC used as measure of model quality

- ARIMA, automated (“forecast” package in R)
  - Best fall model: ARIMA(0,1,0)
  - Best summer model: ARIMA(1,1,0)
  - Best spring model: ARIMA(0,1,0)

Conclusion

- Automated model selection provides rigorous search through all parameter combinations
- Stepping through model-construction process an instructive exercise
  - Offers better understanding of mathematical influences
- Computers best-suited for tasks involving speed and iteration
- Humans best-suited for questions of “why”, application of computer-generated solutions

Thank you!

Questions can be addressed to nyssabulkes at gmail dot com

Thank you to my mentor, Eduardo Carrasco Jr., for guidance along the way, and to Springboard for the opportunity to learn!