

Course Information and Overview

STAT3009 Recommender Systems

by **Ben Dai** (CUHK-STAT)

on **September 4, 2025**

» Course Overview and Key Features

- * **Contemporary** and **Emerging** Field
 - * Recommender systems emerged from industry needs around 2000
 - * This course covers cutting-edge methods from **2010 to present**
 - * Focus on modern deep learning and machine learning approaches
- * **Interdisciplinary** Approach with **Hands-on** Learning
 - * **Theory**: Statistics + Machine Learning + Mathematical Optimization
 - * **Practice**: Python Programming + Real-world Applications
 - * **Delivery**: Interactive lectures + Live Kaggle competitions + Jupyter notebooks

» Course Overview and Key Features

- * **Highly Relevant** to Industry Applications
 - * **E-commerce:** Amazon, Alibaba product recommendations
 - * **Entertainment:** Netflix, Spotify content suggestions
 - * **Social Media:** Facebook, LinkedIn connection recommendations
 - * **Finance:** Investment portfolio recommendations
- * **Distinctive** from Traditional Statistics Courses
 - * Emphasis on computational implementation and programming skills
 - * Strong mathematical foundation required but practical coding equally important
 - * Real-time problem-solving with large-scale datasets

» Course Information and Prerequisites

* **Course Position** in Curriculum

- * STAT3009 is an **elective course** for Statistics majors
- * Also open to students from related quantitative disciplines
- * Bridges theoretical statistics with practical machine learning applications

» Course Information and Prerequisites

* **Course Position** in Curriculum

- * STAT3009 is an **elective course** for Statistics majors
- * Also open to students from related quantitative disciplines
- * Bridges theoretical statistics with practical machine learning applications

* **Prerequisites** and Required Background

- * **Statistical Methods:** Linear regression, ridge regression, basic hypothesis testing
- * **Programming Skills:** Python (NumPy, Pandas), basic scikit-learn usage
- * **Mathematics:** Linear algebra, calculus, probability theory
- * **Note:** Comprehensive tutorials provided for Python libraries

» Class Composition

* **Class Composition** and Diversity

- * **Enrollment:** 49/60 students (as of September 1)
- * **Backgrounds:** STAT, Risk Management, Quantitative Finance, Natural Sciences, CS, Math, ...

Hybrid interdisciplinary + teaching mode

Mix **Statistics** + **Machine Learning** + **Python** + **Mathematics**

Teaching mode

- * lecture slides/note
- * real time Python programming in **colab** (**Jupyter** notebook)
- * InClass practice
- * real time competitions in **Kaggle**

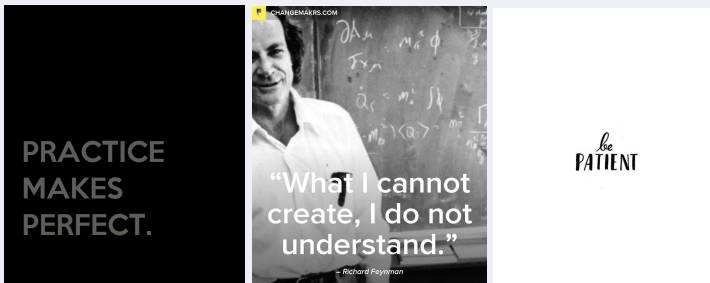
* **Additional Resources**

- * Course Homepage with lecture materials
- * GitHub Repository with code examples

» Grading policy

- * Homework (15%)
- * Inclass Kaggle Competition (Open-book InClass Kaggle Competition) (50%)
- * Final InClass Quiz (coding and exercise) (35%): Basic Python programming and implementation of recommender systems models (during the final lecture of the semester)

» My suggestion



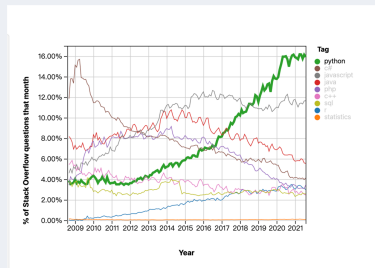
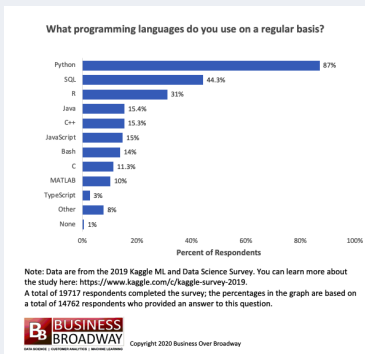
- * **practice** by yourself
 - * enjoy **debugging**: most coding experiences are from that
 - * **independent**: 99.9% bugs can be solved by Google
 - * **bad example**. Just look at the code, but never practice/implement on your own
- You must have a laptop

» Preparation

Toolbox we will use:

Python Python is the programming language of choice for data scientists.

* State of Data Science 2021: **Popularity** of Python



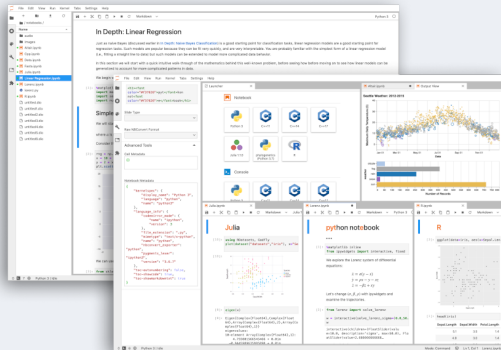
Source: <https://businessoverbroadway.com>

Source: <https://www.alibabacloud.com/blog>

» Preparation

Toolbox we will use:

NB **Jupyter notebook** - create and share documents that contain live code, equations, visualizations and narrative text

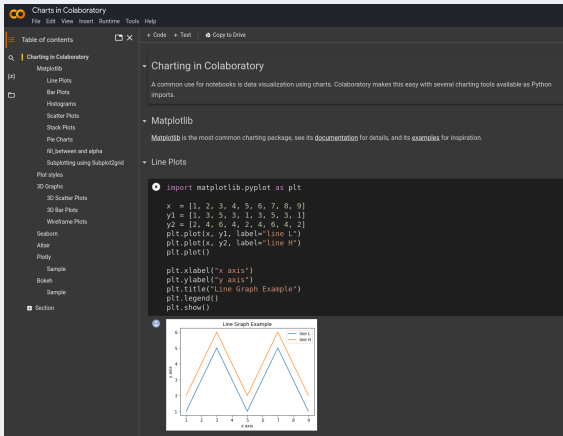


Source: <https://jupyter.org/>

» Preparation

Toolbox we will use:
Colab **Register** - Colaboratory (Google account)

- * Online Python computing platform
- * Python **Jupyter notebook**



The screenshot displays the Colaboratory web interface. On the left is a sidebar with a 'Table of contents' and a search bar. Below these are categories of charts: Matplotlib (Line Plots, Bar Plots, Histograms, Scatter Plots, Stock Plots, Pie Charts, fill_between and alpha, Subplotting using Subplot2grid), Plot styles (3D Graphs, 3D Scatter Plots, 3D Bar Plots, Wireframe Plots), Seaborn, Altair, Plotly, and a 'Section' dropdown. The main area shows a notebook titled 'Charts in Colaboratory'. It contains a code cell with Python code for plotting two lines. Below the code is a preview of the resulting line graph.

Charts in Colaboratory
File Edit View Insert Runtime Tools Help

Table of contents

Charting in Colaboratory

Matplotlib

- Line Plots
- Bar Plots
- Histograms
- Scatter Plots
- Stock Plots
- Pie Charts
- fill_between and alpha
- Subplotting using Subplot2grid

Plot styles

- 3D Graphs
- 3D Scatter Plots
- 3D Bar Plots
- Wireframe Plots

Seaborn

Altair

Plotly

Sample

Book

Sample

Section

Charting in Colaboratory

A common use for notebooks is data visualization using charts. Colaboratory makes this easy with several charting tools available as Python imports.

Matplotlib

Matplotlib is the most common charting package, see its [documentation](#) for details, and its [examples](#) for inspiration.

Line Plots

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
y1 = [1, 3, 5, 3, 1, 3, 5, 3, 1]
y2 = [2, 4, 6, 4, 2, 4, 6, 4, 2]
plt.plot(x, y1, label='line 1')
plt.plot(x, y2, label='line 2')
plt.plot()

plt.xlabel('x axis')
plt.ylabel('y axis')
plt.title('Line Graph Example')
plt.legend()
plt.show()
```

Line Graph Example

The graph shows two lines: 'line 1' (blue) and 'line 2' (orange). Both lines follow a similar pattern, peaking at x=3 and x=7, and reaching a minimum at x=5. The y-axis ranges from 1 to 6, and the x-axis ranges from 1 to 9.

» Preparation

Toolbox we will use:

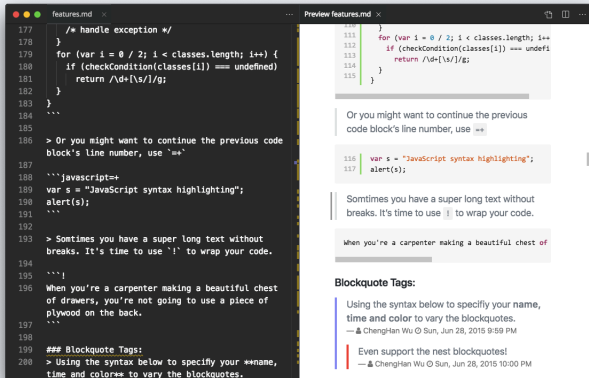
Colab

- * Go to `colab.research.google.com` and sign in with your Google account.
- * Create a new notebook by clicking on the "New Notebook" button.
- * Write Python code in the cells of the notebook, using the same syntax and semantics as you would in a local Python environment.
- * Execute the code by clicking on the "Run" button or pressing Shift+Enter.
- * See the output of your code in the notebook, including any plots, tables, or text output.

» Preparation

Toolbox we will use:

Colab Online **Markdown** documentation



```
177  /* handle exception */
178  }
179  for (var i = 0 / 2; i < classes.length; i++) {
180    if (checkCondition(classes[i]) === undefined)
181      return /\d+[\s]/g;
182  }
183  }
184  ...
185
186  > Or you might want to continue the previous code
    block's line number, use `==`
187
188  ```javascript==
189  var s = "JavaScript syntax highlighting";
190  alert(s);
191  ```
192
193  > Sometimes you have a super long text without
    breaks. It's time to use `!` to wrap your code.
194
195  ```!
196  When you're a carpenter making a beautiful chest of
    drawers, you're not going to use a piece of
    plywood on the back.
197  ```
198
199  ### Blockquote Tags:
200  > Using the syntax below to specify your **name,
    time and color** to vary the blockquotes.
```

```
111  }
112  for (var i = 0 / 2; i < classes.length; i++)
113    if (checkCondition(classes[i]) === undefi
114      return /\d+[\s]/g;
115  }
```

Or you might want to continue the previous code block's line number, use `==`

```
116  var s = "JavaScript syntax highlighting";
117  alert(s);
```

Sometimes you have a super long text without breaks. It's time to use `!` to wrap your code.

When you're a carpenter making a beautiful chest of

Blockquote Tags:

Using the syntax below to specify your name, time and color to vary the blockquotes.

— ChengHan Wu ☉ Sun, Jun 28, 2015 9:59 PM

Even support the nest blockquotes!

— ChengHan Wu ☉ Sun, Jun 28, 2015 10:00 PM

» Preparation

Toolbox we will use:
Kaggle - online community of data scientists and machine learning practitioners

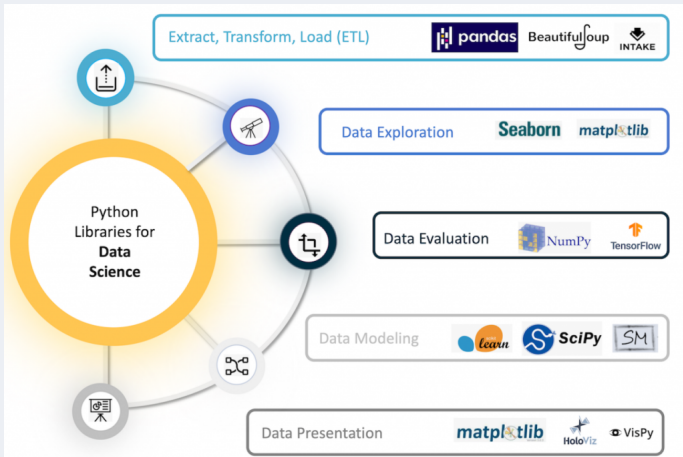
The screenshot displays the Kaggle homepage with a navigation bar at the top containing links for All Competitions, Featured, Getting Started, Research, Community, and Playground. Below the navigation bar, the 'Active Competitions' section is highlighted, showing a grid of 12 competitions. Each competition card includes a thumbnail image, the competition title, a brief description, the number of featured teams, the prize amount, and the time remaining.

Competition	Prize	Time to go
1st and Future - Player Contact Detection	\$100,000	3 months to go
NFL Big Data Bowl 2023	\$100,000	a month to go
Santa 2022 - The Christmas Card Conundrum	\$50,000	a month to go
RSNA Screening Mammography Breast Canc...	\$50,000	2 months to go
OTTO - Multi-Objective Recommender System	\$30,000	2 months to go
Novozymes Enzyme Stability Prediction	\$25,000	22 days to go
G2Net Detecting Continuous Gravitational Waves	\$25,000	22 days to go
Scrabble Player Rating	Kudos	3 days to go
Lux AI 2022 - Beta		
Predict Future Sales		
Titanic - Machine Learning from Disaster		
House Prices - Advanced Regression Techniques		

» Preparation

Toolbox we will use:

Libs Python libraries for Data Science



» Preparation

- * Python package installation

- * Basic packages: numpy + pandas + seaborn + scipy + scikit-learn
- * Install packages in Jupyter notebook / colab



install python package

```
!pip install <package name>
```