

# HW\_ClassSearch

October 7, 2024

## 1 Name: Ramya Chowdary Patchala

## 2 PRACTICE - iSchool Class Search

### 2.1 The Problem

You have been hired to build an interactive data product for the iSchool that makes it easier for students to find classes. Your task is to read in a schedule of classes and create a user interface that allows someone to search the classes by:

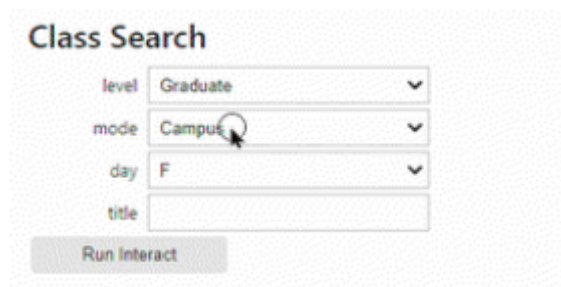
- Level: Graduate / Undergraduate
- Mode: Campus / Online
- Meeting: MW / TuTh / MWF / etc...
- Contents of the Course title

The program should then output a dataframe of the courses which match the selected criteria.

If this were a real-world project there would be two key steps - building the data pipeline to acquire the necessary data and then - building the user interface around the data.

The data can be found at the following URL: <https://raw.githubusercontent.com/mafudge/datasets/master/>

See the final product layout here:



### 2.2 Approach:

This assignment is broken up into parts. We will use problem simplification to solve this problem and take a bottom up approach, making the components, then assembling them together.

- Load the data into a pandas dataframe
- Data Cleanup
- Engineer the Columns we Need

- Building data for our widgets
- Assemble the final program from its parts

## 2.3 Part 1: Code Solution

You may write your code in several cells, but place the complete, final working copy of your code solution.

### 2.3.1 You Code 2.1: Load the data into a pandas dataframe

In this first step, import the `pandas` and `numpy` libraries and then write code to load the dataset from the url found in the instructions at the top. Load into a Pandas DataFrame.

Also for your own sanity, you should probably ignore Pandas `filterwarnings`. We did this in the lab.

use the `print()` function to display the first 10 classes. The code checker will scan this output so if you want this to pass you will need to use `print()` instead of `display()`

Just know that you can use `display()` while figuring it out, but if you want to pass the code checks, you'll have to switch to `print()`

```
[3]: # SOLUTION CELL 2.1
import pandas as pd
import numpy as np
import warnings

# Ignore Pandas filter warnings
warnings.filterwarnings('ignore')

# Load data from the provided URL
url = "https://raw.githubusercontent.com/mafudge/datasets/master/classes/
      ↪ischool-schedule-fall2015.csv"
class_df = pd.read_csv(url)

# PRINT FIRST 10 ROWS
print(class_df.head(10))
```

	Course	Section	Class	Credits	Title \
0	IST990	M001	29957	1.0	Independent Study
1	IST999	M001	21631	1.0	Dissertation
2	IST625	M001	21661	3.0	Enterprise Risk Management
3	IST625	M002	21902	3.0	Enterprise Risk Management
4	IST627	M002	21755	3.0	What's the Big Idea
5	IST639	M001	21696	3.0	Enterprise Technologies
6	IST641	M001	21711	3.0	User-Based Design
7	IST645	M001	21664	3.0	Managing Info Systems Projects
8	IST645	M002	21665	3.0	Managing Info Systems Projects
9	IST645	M800	21666	3.0	Managing Info Systems Projects

	Instructor(s)	Time	Day \
0	NaN	12:00am - 12:00am	NaN
1	NaN	12:00am - 12:00am	NaN
2	Michelle L. Brown	9:30am - 12:15pm	W
3	Frank Jr Marullo	5:00pm - 7:50pm	Tu
4	William C Padgett Marcene S. Sonneborn	2:00pm - 3:20pm	TuTh
5	P Douglas Taber	5:15pm - 8:05pm	W
6	Michael S Nilan	12:30pm - 3:15pm	Th
7	Arthur P. Thomas	5:00pm - 7:45pm	Tu
8	Tom Uva	5:15pm - 8:05pm	M
9	Robert A Emborski	12:00am - 12:00am	NaN

	Room(s)
0	NaN
1	NaN
2	Hinds Hall 021
3	Newhouse 1 101
4	Hinds Hall 021
5	Hinds Hall 027 Hinds Hall 111
6	Hinds Hall 120
7	Hinds Hall 021
8	Hinds Hall 117
9	Online

### 2.3.2 You Code 2.2: Data Cleanup

If you look over the data with `info()` you will notice there are missing values in the `Instructor(s)`, `Day`, and `Room(s)` columns. We need to clean this data up before presenting it as the missing values showing "NaN" will be confusing to the users of our program.

Specifically do the following:

- in the `Instructor(s)` column replace all NaN with "Staff". Its common for universities to use this label when the instructor is to be determined.
- in the `Room(s)` column replace NaN with "TBA". Its common for universities to use this label when the room will be announced later TBA == To be Announced.
- in the `Day` column replace NaN with "N/A". N/A means not applicable.

TIPS :

- use the column-selector then boolean filter approach : `df[col][boolean-index-selector] = value`
- your boolean index selector cannot compare the column to `np.nan` e.g. `col == np.nan` this is not the way to find nulls in a series!
- if you screw up your dataframe, don't fret just run 2.1 to reload it!

```
[11]: # SOLUTION CELL 2.2

# COPY CODE FROM 2.1 INCLUDE THE IMPORTS!
import pandas as pd
```

```

import numpy as np
import warnings

# Ignore Pandas filter warnings
warnings.filterwarnings('ignore')

# Load data from the provided URL
url = "https://raw.githubusercontent.com/mafudge/datasets/master/classes/
school-schedule-fall2015.csv"
class_df = pd.read_csv(url)

# YOUR CLEANUP CODE
class_df['Instructor(s)'].fillna('Staff', inplace=True)
class_df['Room(s)'].fillna('TBA', inplace=True)
class_df['Day'].fillna('N/A', inplace=True)

# FOR CHECKER: PRINT ROWS
print(class_df.head(10))

```

	Course	Section	Class	Credits	Title \
0	IST990	M001	29957	1.0	Independent Study
1	IST999	M001	21631	1.0	Dissertation
2	IST625	M001	21661	3.0	Enterprise Risk Management
3	IST625	M002	21902	3.0	Enterprise Risk Management
4	IST627	M002	21755	3.0	What's the Big Idea
5	IST639	M001	21696	3.0	Enterprise Technologies
6	IST641	M001	21711	3.0	User-Based Design
7	IST645	M001	21664	3.0	Managing Info Systems Projects
8	IST645	M002	21665	3.0	Managing Info Systems Projects
9	IST645	M800	21666	3.0	Managing Info Systems Projects

	Instructor(s)	Time	Day \
0	Staff	12:00am - 12:00am	N/A
1	Staff	12:00am - 12:00am	N/A
2	Michelle L. Brown	9:30am - 12:15pm	W
3	Frank Jr Marullo	5:00pm - 7:50pm	Tu
4	William C Padgett Marcene S. Sonneborn	2:00pm - 3:20pm	TuTh
5	P Douglas Taber	5:15pm - 8:05pm	W
6	Michael S Nilan	12:30pm - 3:15pm	Th
7	Arthur P. Thomas	5:00pm - 7:45pm	Tu
8	Tom Uva	5:15pm - 8:05pm	M
9	Robert A Emborski	12:00am - 12:00am	N/A

	Room(s)
0	TBA
1	TBA

```

2           Hinds Hall 021
3           Newhouse 1 101
4           Hinds Hall 021
5 Hinds Hall 027 Hinds Hall 111
6           Hinds Hall 120
7           Hinds Hall 021
8           Hinds Hall 117
9           Online

```

### 2.3.3 You Code 2.3 - Engineer the columns we need

Next we need to engineer two columns. Here are the criteria:

- Column name `Level`, value is:
  - "Graduate" number part of the course is  $\geq 500$  , e.g. IST625 ( $625 \geq 500$ )
  - "Undergraduate" number part of the course is  $< 500$  e.g. IST256 ( $256 < 500$ )
- 
- Column name `Mode`, value is:
  - "Online" 2nd character in Section is an "8" e.g. M800
  - "Campus" 2nd character in Section is not an "8" e.g. M012

TIPS :

- Again, use the column-selector then boolean filter: `df[col][boolean-index-selector] = value`

```

[12]: # SOLUTION CELL 2.3

# COPY CODE FROM 2.1 INCLUDE THE IMPORTS!
# SOLUTION CELL 2.1
import pandas as pd
import numpy as np
import warnings

# Ignore Pandas filter warnings
warnings.filterwarnings('ignore')

# Load data from the provided URL
url = "https://raw.githubusercontent.com/mafudge/datasets/master/classes/
      ↪ischool-schedule-fall2015.csv"
class_df = pd.read_csv(url)

# ENGINEER COLUMNS
class_df['Level'] = class_df['Course'].str[3:]
# Convert to Integer
class_df['Level'] = class_df['Level'].astype(int)
# Classify Courses using the sliced numeric part

```

```

class_df['Level'] = np.where(class_df['Level'] >= 500, 'Graduate',
    ↪ 'Undergraduate')

class_df['Mode'] = np.where(class_df['Section'].str[1] == '8', 'Online',
    ↪ 'Campus')

# FOR CHECKER: PRINT SLICE FROM 100 to 106
print(class_df.iloc[100:107, [0, 1, -2, -1]])

```

	Course	Section	Level	Mode
100	IST623	M001	Graduate	Campus
101	IST623	M002	Graduate	Campus
102	IST323	M002	Undergraduate	Campus
103	IST323	M003	Undergraduate	Campus
104	IST323	M004	Undergraduate	Campus
105	IST325	M001	Undergraduate	Campus
106	IST335	M001	Undergraduate	Campus

### 2.3.4 You Code 2.4 Building data for our widgets

Next we need to build the data for our dropdown input widgets. We need three:

1. a sorted list of unique non NaN values in the Mode Series, call this variable `modes`
2. a sorted list of unique non NaN values in the Level Series, call this variable `levels`
3. a sorted list of unique non NaN values in the Day Series, call this variable `days`

TIPS :

- Take the approach we used in the homework.
- You do not need to create a custom widget here, just the Python lists of unique values.

```

[7]: # SOLUTION CELL 2.4

# COPY CODE FROM 2.1, 2.2 and 2.3

import pandas as pd
import numpy as np
import warnings

# Ignore Pandas filter warnings
warnings.filterwarnings('ignore')

# Load data from the provided URL
url = "https://raw.githubusercontent.com/mafudge/datasets/master/classes/
    ↪ ischool-schedule-fall2015.csv"
class_df = pd.read_csv(url)

class_df['Instructor(s)'].fillna('Staff', inplace=True)

```

```

class_df['Room(s)'].fillna('TBA', inplace=True)
class_df['Day'].fillna('N/A', inplace=True)

class_df['Level'] = np.where(class_df['Course'].str.extract('(\d+')[0].
    ↪astype(int) >= 500, 'Graduate', 'Undergraduate')

class_df['Mode'] = np.where(class_df['Section'].str[1] == '8', 'Online',
    ↪'Campus')

# CREATE LISTS
modes = sorted(class_df['Mode'].unique())
levels = sorted(class_df['Level'].unique())
days = sorted(class_df['Day'].unique())

# FOR CHECKER: PRINT EACH LIST
print("Modes:", modes)
print("Levels:", levels)
print("Days:", days)

```

```

Modes: ['Campus', 'Online']
Levels: ['Graduate', 'Undergraduate']
Days: ['F', 'M', 'MW', 'MWF', 'N/A', 'SaSu', 'Th', 'Tu', 'TuTh', 'W', 'WF']

```

### 2.3.5 You Code 2.5 Assemble the final program as an interact

With all the components built, its time to consider the complete program.

TIPS:

- As you write your algorithm remember you will perform steps 2.1 - 2.4 **before** you accept any input.
- Since there are 4 inputs, there should be 4 arguments to `@interact_manual` and the `on_click()` function.
- use `display()` to print the filtered dataframe.

```

[13]: # SOLUTION CELL 2.5
import pandas as pd
import numpy as np
import warnings
from IPython.display import display, HTML
from ipywidgets import interact_manual
warnings.filterwarnings('ignore')

# Function to filter df
def filter_classes(level, mode, day, title):
    filtered_df = class_df[
        (class_df['Level'] == level) &

```

```

        (class_df['Mode'] == mode) &
        (class_df['Day'].str.contains(day, na=False)) &
        (class_df['Title'].str.contains(title, na=False, case=False))
    ]
    display(filtered_df)

# Create the interactive widget
interact_manual(
    filter_classes,
    level=levels,
    mode=modes,
    day=days,
    title=""
)

```

```

interactive(children=(Dropdown(description='level', options=('Graduate', 'Undergraduate'), value='Graduate'), ...

```

```
[13]: <function __main__.filter_classes(level, mode, day, title)>
```

## 2.4 Part 3: Metacognition

These questions are designed to prompt you to reflect on your learning. Reflection is part of the assignment grade so please take time to answer the questions thoughtfully.

### 3.1 List at least 3 things you learned this week and/or throughout the process of completing this assignment?

1. Working with dataframes in pandas
2. Exploring and visualizing in pandas
3. Preparing interactive widgets

### 3.2 What were the challenges or roadblocks (if any) you encountered on the way to completing it? No

### 3.3 Were you prepared for this assignment? What can you do to be better prepared? Yes

### 3.4 Did someone (or something such as AI) help you? Did You help someone? Provide details. No

**3.5 Now that you have completed the assignment rate your comfort level with this week's material. This should be an honest assessment of your ability:** 1 ==> I don't understand this at all yet and need extra help. If you choose this please try to articulate that which you do not understand to the best of your ability in the questions and comments section below.  
 2 ==> I can do this with help or guidance from other people or resources. If you choose this level, please indicate HOW this person helped you in the questions and comments section below.



**3** ==> I can do this on my own without any help.

**4** ==> I can do this on my own and can explain/teach how to do it to others.

ENTER A NUMBER 1-4 IN THE CELL BELOW

4