Lesson Plan for CIS 3100 Fall 2023

Month	Day	Demonstration topic	Assignment number: title	Homework, assessment on MyLab	videos
aug	**24**	Introduction day			
	29	IT infrastructure and application development skills	Video here		1
	31	Notebook basics, python basics		1	here
sep	**5,7**	Websocket dev	1: Basic IT Infrastructures	2	Venv , cbpro
	12,14, 19	Websocket manip, functions	2: Basic Applications	3	Part 1, part 2
	21	Csv output and joins into one dataframe	3: Basic Data Preparation for Analytics	4	Part 1, part 2
	26,28	Excel / Business Intelligence	4. Business Intelligence	5	Part 1, part 2, part 3
oct	**3,5**			6	Part 4, part 5
	10,12	Work on our tool assignment (excel/BI)	Wrap up our excel BI/tool assignment	7	
	17,19		5: Basic Data Science, notebook python	8	Part 1, assignment here
	24,26, 31		6: Basic Data Science 2, the Ulta project redux	9	part 2, part 3, part 4 wrap up
	2	Communicating Data Science Results	7: github and online collaboration	10	Part 1
nov	**7,9, 14**	Machine Learning 1: Data Engineering	8: Basic Supervised Learning, Data Engineering and Applications		Project start, video one, video two, video three
	16	Machine Learning 2: Predictive Analytics	9: Machine Learning 2: Unsupervised Learning		Intro video here
	21				Video here
	28,30	Python solve for excel project	10: Applied Machine Learning Project		Video 1,
dec	**5,7**	*Spare days, open budget*			
	12,14	12/18 noon		Please note	

Month	Day	Demonstration topic	Assignment number: title	Homework, assessment on MyLab	videos
aug	**24**	Introduction day			
	29	IT infrastructure and application development skills	Video here		1
	31	Notebook basics, python basics		1	here
sep	**5,7**	Websocket dev	1: Basic IT Infrastructures	2	Venv , cbpro
				that my lab times are in the eastern time zone and are three hours ahead.	

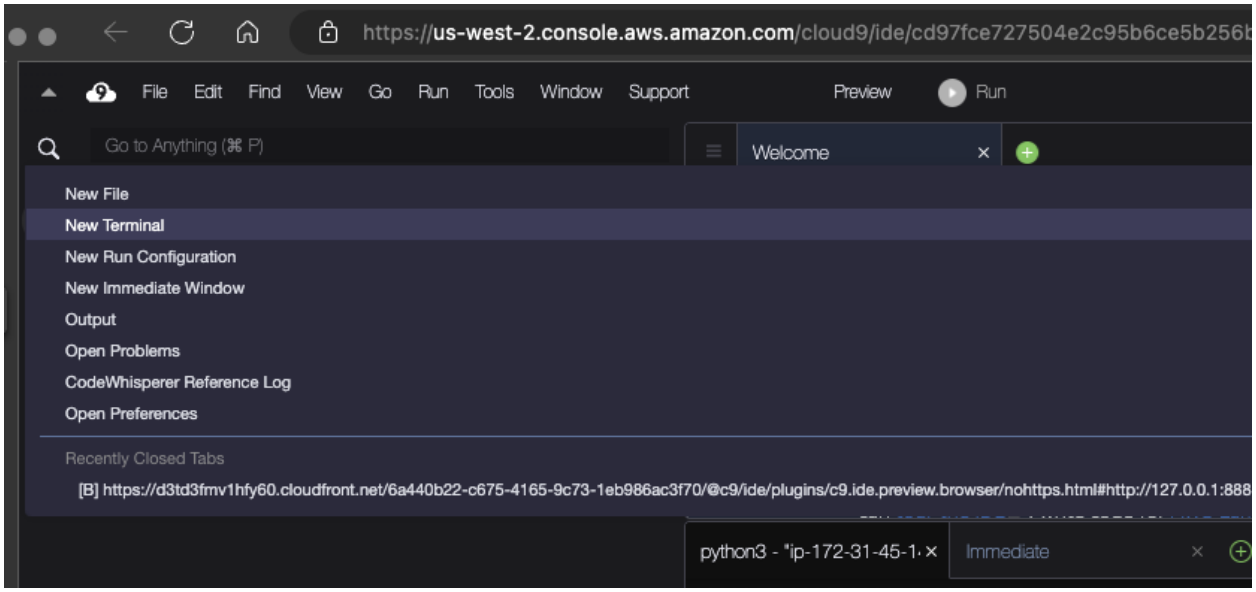
Introductory

Opening talk, video
Video 2, day 2
Video 3, day 3

1: Basic IT Infrastructures

Aws setup, ide setup

Establish a terminal session like so:



install python, for the correct version
[Your Guide to pyenv | LearnPython.com]

```
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt update
sudo apt install python3.8
sudo apt install python3.8-venv
python3.8 -m venv fall3100
ls
cd fa*
 ls
cd bin
 ls
 source activate
 pip3 install cbpro
pip3 install pandas
pip3 install jupyter
Cd ..
Cd ..
```

Source Code to try during the experience

Code starts here:

```
import csv
import cbpro
import numpy as np
```

```
public_client = cbpro.PublicClient()
bk = public_client.get_product_order_book('BTC-USD', level=3)
print("book type:",bk)
```

Project 2: Basic Applications

Original:
https://github.com/stefanbund/grus-code/blob/main/obaTrader/obaMain-bidListEnabled.py

Our code:

```
import csv
import cbpro
import numpy as np

import pandas as pd


public_client = cbpro.PublicClient()

book = public_client.get_product_order_book('BTC-USD', level=3)
RANGE = 0.0036
# print("book type:",bk)
def sum_qty_asks(df):
    df['price'] = df['price'].astype(float)
    df['qty'] = df['qty'].astype(float)
    top_price = float(df['price'].min())
    threshold = top_price * RANGE
    return df[df['price'] >= top_price - threshold]['qty'].sum()

def sum_qty_bids(df):
    df['price'] = df['price'].astype(float)
    df['qty'] = df['qty'].astype(float)
    top_price = float(df['price'].max())
    threshold = top_price * RANGE
    return df[df['price'] >= top_price - threshold]['qty'].sum()

def sum_bids_cap(df):
```

```
    df['price'] = df['price'].astype(float)
    df['qty'] = df['qty'].astype(float)
    top_price = float(df['price'].max())
    threshold = top_price * RANGE
    df['sum']= df['price'] * df['qty']
    caps = df[df['price'] >= top_price - threshold]['sum'].sum()
    return caps

def sum_asks_cap(df):
    df['price'] = df['price'].astype(float)
    df['qty'] = df['qty'].astype(float)
    top_price = float(df['price'].min())
    threshold = top_price * RANGE
    df['sum']= df['price'] * df['qty']
    caps = df[df['price'] >= top_price - threshold]['sum'].sum()
    return caps
ba = book['asks'] #book of asks
bb = book['bids'] #book of bids

df_asks = pd.DataFrame(ba, columns=['price','qty','id'])

df_bids = pd.DataFrame(bb, columns=['price','qty','id'])

print("ask qty", df_asks.shape[0])

print("bid qty" ,df_bids.shape[0] )

print(df_asks.columns)

print(df_asks['price'].head(25))

tav = round(sum_qty_asks(df_asks),3)
tbv = round(sum_qty_bids(df_bids),3)
bc = round(sum_bids_cap(df_bids),3)
ac = round(sum_asks_cap(df_asks),3)
mp = df_asks.iloc[0]['price']

print(tav, tbv, bc, ac, mp)
```

App 2:

```
class Product:
   def __init__(self, product_type, price, volume, brand, title):
```

```
        self.type = product_type  #mascara, facial cream,
        self.price = price   #variables
        self.volume = volume
        self.brand = brand
        self.title = title

class Store:
    def __init__(self):
        self.products = []  #array

    def add_product(self, product_type, price, volume, brand, title):
        new_product = Product(product_type, price, volume, brand, title)
        self.products.append(new_product)

    def print_products(self):
        print("Products:")
        for i, product in enumerate(self.products):
            print(f"{i+1}. Type: {product.type}, \
            Price: {product.price}, \
            Volume: {product.volume},\
            Brand: {product.brand}, \
            Title: {product.title}")


my_store = Store()                    #first variable

my_store.add_product("eye liner", 50.0, "10ml", "Maybelline", "Age Defying Mist")
my_store.add_product("eye liner", 40.0, "13ml", "Benefit", "Fountain of Youth")
my_store.add_product("eye liner", 30.0, "8ml", "Loreal", "Start it off with a kiss")


my_store.print_products() # Print all the products in the store
```

To submit:

```
python basic.py >> store.txt
```

Project 3

```
import csv
import cbpro
import numpy as np
import pandas as pd
```

```
import sched, time
from datetime import date
public_client = cbpro.PublicClient()

book = public_client.get_product_order_book('BTC-USD', level=3)
RANGE = 0.0036
# print("book type:",bk)
def sum_qty_asks(df):
    df['price'] = df['price'].astype(float)
    df['qty'] = df['qty'].astype(float)
    top_price = float(df['price'].min())
    threshold = top_price * RANGE
    return df[df['price'] >= top_price - threshold]['qty'].sum()

def sum_qty_bids(df):
    df['price'] = df['price'].astype(float)
    df['qty'] = df['qty'].astype(float)
    top_price = float(df['price'].max())
    threshold = top_price * RANGE
    return df[df['price'] >= top_price - threshold]['qty'].sum()

def sum_bids_cap(df):
    df['price'] = df['price'].astype(float)
    df['qty'] = df['qty'].astype(float)
    top_price = float(df['price'].max())
    threshold = top_price * RANGE
    df['sum']= df['price'] * df['qty']
    caps = df[df['price'] >= top_price - threshold]['sum'].sum()
    return caps

def sum_asks_cap(df):
    df['price'] = df['price'].astype(float)
    df['qty'] = df['qty'].astype(float)
    top_price = float(df['price'].min())
    threshold = top_price * RANGE
    df['sum']= df['price'] * df['qty']
    caps = df[df['price'] >= top_price - threshold]['sum'].sum()
    return caps

def generateSummaryonLOB(scheduler):
    scheduler.enter(3, 1, generateSummaryonLOB, (scheduler,))
    book = public_client.get_product_order_book('AVAX-USD', level=3)
    da = book['asks']
    df_asks = pd.DataFrame(da, columns=['price','qty','id'])
    db = book['bids']
```

```
    df_bids = pd.DataFrame(db, columns=['price','qty','id'])

    tav = round(sum_qty_asks(df_asks),3)
    tbv = round(sum_qty_bids(df_bids),3)
    bc = round(sum_bids_cap(df_bids),3)
    ac = round(sum_asks_cap(df_asks),3)
    mp = df_asks.iloc[0]['price']
    writer.writerow([mp, tav, tbv, bc, ac, int(time.time())]) #array
of veriables, not a dictionary

today = date.today()
filename = today.strftime("%Y-%m-%d")+ ".csv"
with open(filename, mode='w', newline='') as csv_file:
    writer = csv.writer(csv_file)
    writer.writerow(['mp', 'tav', 'tbv', 'bc', 'ac', 'time'])   #init
once with headers, mp, tav, tbv, bc, ac, time
    my_scheduler = sched.scheduler(time.time, time.sleep)
    my_scheduler.enter(3, 1, generateSummaryonLOB, (my_scheduler,))
#generateSummary insert
    my_scheduler.run()


#_________________ end copy
```

Ulta kids:

```
import random

class Product:
    def __init__(self, name, cost, volume, title=None):
        self.name = name
        self.cost = cost
        self.volume = volume
        self.title = title

    def __str__(self):
        return f"{self.name} ({self.volume} ml) - ${self.cost:.2f}" +
(f" - {self.title}" if self.title else "")

# List of 300 adjectives and nouns
adjectives = ["moisturizing", "enhanced", "age-defying", "hydrating",
"revitalizing", "soothing", "nourishing", "brightening",
"rejuvenating", "firming", "smoothing", "calming", "illuminating",
```

```
"protective", "balancing", "clarifying", "restorative", "radiant",
"glowing", "refreshing", "strengthening", "volumizing", "softening",
"exfoliating", "pore-minimizing"]
nouns = ["shampoo", "conditioner", "body wash", "lotion", "face
cream", "serum"]

# Generate 400 products with random titles
products = []
for i in range(200000):
    name = f"Product {i+1}"
    cost = random.uniform(1, 50)
    volume = random.randint(100, 500)
    title = f"{random.choice(adjectives)} {random.choice(nouns)}"
    products.append(Product(name, cost, volume, title))

# Print the first 10 products
for product in products:
    print(product)
```

Project 4: Excel Business Intelligence

1. Try these github commands:

```
mkdir receiver
  cd receiver
  ls
  git clone https://github.com/stefanbund/py3100/
  ls
  cd py3100
```

Then

```
    pip3 install pandas
```

[run stuff inside of py3100] per our video instructions

Data production is not required, but go to https://github.com/stefanbund/py3100/ in order to procure your dataset

or
Spreadsheet linked for today, 41 items

Project Assignment
Devise a tool to analyze weekly sales numbers from a major corporate retail enterprise.

Assume these points:

1. Excel will be used
2. Corporate data arrives as a .csv file, such as 'matrix.csv' in our github
3. Pivot Tables are the best tool for the data at hand.
4. Data is in the format of a store, and 40+ retail displays, where many products are displayed from many types of products.
5. The executives are less concerned with the granular details (ie which products are selling). We are interested in how effectively our retail displays are working.
6. Data is expressed as integer values, representing the total sales, by display.

Objectives

1. Work individually **or** as a team of 3, max.
2. You must be able to determine how effectively a retail display creates sales. Assume that a team of designers assembled the layout of each display, and they are accountable for the sales conversion for each display.
3. Deliver a tool that enables the user to select a store, or a group of stores, by city. Then, enable the user to select which display (one or more) to show.
4. Provide separate sheets that display these **graphics** (one per sheet), where a pivot table enables a graph:

    a. The top performing displays for all stores.
    b. The top performing stores, by sales volume, based on all displays
    c. The worst performing displays, across all stores
    d. The worst performing stores, by sales volume, based on all displays.
5. You may make graphs in a style, color and format according to your judgment. Be sure to deliver a graph that is easily understood.
6. You may submit your work as an excel workbook, to canvas.
7. When you create a chart, be sure to label each axis, and provide a descriptive title.

Some notes:

To highlight the cell containing the maximum value in an Excel sheet, you can use **conditional formatting**. Here are the steps to follow:

1. Select the range of cells you want to apply the formatting to.
2. Click on the **Home** tab in the ribbon.
3. Click on **Conditional Formatting** and select **New Rule**.
4. In the **New Formatting Rule** dialog box, select **Use a formula to determine which cells to format**.
5. In the **Format values where this formula is true** field, enter `=A1=MAX($A$1:$A$10)` where `A1` is the first cell in your selected range and `$A$1:$A$10` is the range of cells you want to compare against.
6. Click on **Format** and choose your desired formatting options.
7. Click on **OK** twice to close both dialog boxes.

This will highlight the cell containing the maximum value in your selected range.

Project 5: Business Analytics: Ulta Part 2

3100-03, **4 /7 pm** session Please start with this notebook

3100-05 OL session please use the above notebook

Link to our github: https://github.com/stefanbund/py3100/tree/main

Project 6: Basic Data Science and Visualizing Big Data

Project submission guidelines

1. You will make a copy of Bund's Ulta notebook
2. You will fill in markdown segments for every cell in the notebook, including markdown that is already there.
3. A full description of what transpires in the cell is needed, alongside of a very brief description of the necessity of each cell. You should write one to four sentences per cell.
4. Save the notebook as a pdf and upload to canvas.

Project 7: github

Please perform the steps in our video, above, then submit the URL to your github repository, to canvas.

Project 8: Introductory ML: Supervised Learning / Classification

Notebook 1 link

Project Instructions

1. Please copy the notebook to your google drive
2. For each cell, provide one to four sentences, summarizing what was done in the cell. Utilize verbiage and concepts we present in the final lecture for the project (listed above)
3. "Print" the notebook, then save it as pdf, then upload the pdf to canvas. The final version will contain Bund's code, and your explanation as a readable document.

Notes

You might upload this and notebooks for project 6 to your github!

9: Machine Learning 2: Unsupervised Learning

Our work will focus on this google notebook, here.

To complete it, you must apply your own symbol_dict, and save as a pdf.

10. Applied Unsupervised Learning Project

To complete this, you must write markdown in each cell where Stefan wrote his discussions. Interpret the data as you observe it. You should deliver several observations based on your notebook experiment. Please preserve any bibliographic references, and do not delete them.