**Economics 1680: Machine Learning, Text Analysis, and Economics**

Brown University

Prof. Amy Handlan

In this course we will be using Python for writing code to apply machine learning and text analysis methods to economics topics. Python is free, flexible, offers a variety of predefined packages, and is popular. It can handle everything from the statistical analysis of Stata to the matrix algebra and simulation of Matlab.

This assignment is meant to introduce you to how we will be using Python in this course. For this assignment, you should write/type your answers into this worksheet. You may discuss the problem set with your class mates, but every student must do their own work.

It is always important to cite our references that help us in our work. Please cite the students you work with here: \_\_\_ , \_\_\_\_, \_\_\_\_\_

1. **PRELIMINARIES (20 points total)**

* Download and install Python/Anaconda onto your computer:

First, you need to install python 3.x and packages for scientific computation on your computer. I recommend you download and install the Anaconda distribution of python because it comes with the majority of packages you will need. An additional bonus of working with Anaconda is it has multiple environments for editing and running python code, including Spyder and Jupyter. Go here to begin downloading the individual edition of Anaconda (Windows, Mac, Linux): <https://www.anaconda.com/products/individual>

Students are expected to work in python for this class. If you need a refresher on the language, here are additional resources for learning python:

* http://www.codecademy.com/en/tracks/python
* http://www.learnpython.org
* The book “Learning Python” by Mark Lutz

I *recommend* using Spyder or Jupyter so that you can edit and run python code in an *interactive* environment on your computer.

* Packages to install for this assignment
  + You will need to install the Nasdaq data link, nltk, and wordcloud using conda, pip, or pip3 install in the terminal for Mac or Linux computers, and the Anaconda Prompt for Windows computers. If not using Anaconda, you can use pip install for these packages.

|  |
| --- |
| conda install -c conda-forge nasdaq-data-link  conda install -c anaconda nltk  conda install -c conda-forge wordcloud |

* Set up Github account and connect to Github Classroom for Econ 1680:

For the assignments in this class you will be expected to submit your code into a git repository in our Github Classroom. Github is a website that allows coders to manage, store, track, and share code. Github is both helps with building good coding habits and a great way to showcase your code for future employers and institutions. Your project files and assignments will be stored in a private repository for Econ 1680 that only you, the professor, and the teaching assistant can see. A repository contains your files and each file’s history.

* From Homework Assignment 1 page in Canvas, click on the HW 1 invitation link to access the course GitHub repository. You will be prompted to connect to the Github Classroom.
* Link your GitHub account to the classroom and upload your final HW 1 code to the Github Classroom.
* Either log into your GitHub Account or set up a free account on GitHub. You can sign up for an account at: <https://github.com/signup>. When you click into the Github classroom, you will be prompted to log in to your account to link your Github Account to your identifier in the Econ 1680 Github classroom. Note: we will be able to see your linked username, so please make sure your Github username is appropriate and professional.
* If you are new to using GitHub or commands in the Terminal Prompt, I recommend downloading the GitHub Desktop app. This will allow you to clone git repositories, submit your homework, and practice using the git workflow in a more interactive way.
* In the GitHub Classroom, there are instructions on how to submit your homework assignment and code once you are done.

1. **NUMERICAL DATA (40 points total)**
2. Access Zillow Real Estate Data using the Nasdaq Data Link API. Nasdaq Data Link is a dataset aggregation website that also has other economics datasets. These types of websites can make it easier to get data and to explore what types of datasets are available.
   1. Set up free account with Nasdaq Data Link (<https://data.nasdaq.com/>). Find your API Key in your Account Settings. You will need this to download the data.
   2. Find the “Zillow Real Estate Data” that is Free (<https://data.nasdaq.com/databases/ZILLOW/data>) This will be the data you will download.
   3. Click on the “Usage” tab, then select the “Python” sub-tab for instructions on using the Nasdaq Data Link API.
   4. To decide which variables and regions we want to download data for, we will first download information on the indicators and regions. In a python environment, you will run the following to import packages, setup your API connection, and download the indicator and region dataframes:

|  |
| --- |
| ﻿import nasdaqdatalink  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  nasdaqdatalink.ApiConfig.api\_key = 'YOUR\_API\_KEY'  ﻿df\_zillow\_indicators = nasdaqdatalink.get\_table('ZILLOW/INDICATORS' ﻿,paginate=True)  ﻿df\_zillow\_regions = nasdaqdatalink.get\_table('ZILLOW/REGIONS' ﻿,paginate=True) |

* + 1. What does ZVHI in the indicator descriptions stand for? (1 point)

|  |
| --- |
| Answer: |

* + 1. What is the indicator, description, and category of row 38 in ﻿df\_zillow\_indicators? Hint: use .iloc[] (2 points)

|  |
| --- |
| Answer: |

* + 1. In df\_zillow\_regions, how many regions are there when you search for “Providence; RI”? What is the region\_id number for Providence, RI? Hint: use .str.contains(‘Providence; RI’) (4 points)

|  |
| --- |
| Answer: |

* + 1. Download a dataframe the city of Providence, RI on ZHVI Single-Family Home values with the correct indicator and region IDs entered:

|  |
| --- |
| ﻿df\_zillow\_sfh= nasdaqdatalink.get\_table('ZILLOW/DATA', indicator\_id=' ' , region\_id=' ',paginate=True) |

1. Descriptive statistics
   1. What is the data frequency in df\_zillow\_sfh? (1 points)

|  |
| --- |
| Answer: |

* 1. What is the median dollar value of a home in df\_zillow\_sfh? (4 points)

|  |
| --- |
| Answer: |

* 1. What is the median dollar value of a home in df\_zillow\_sfh for the year of 2020? Hint: use ﻿[df\_zillow\_sfh['date'].dt.year==2020] (4 points)

|  |
| --- |
| Answer: |

1. Visualize the Data
   1. Plot a time series graph for values df\_zillow\_sfh. Be sure to title your graph and label your axes. (7 points)

|  |
| --- |
| <Insert Graph Here> |

* 1. Plot time series graph for *yearly median* values df\_zillow\_sfh. Be sure to title your graph and label your axes. Hint: you will can create a new dataframe by creating a ‘year’ column using .dt.year and then use .groupby(by=[‘year’]).median() to make a yearly dataframe. (10 points)

|  |
| --- |
| <Insert Graph Here> |

* 1. What looks different in these graphs? Why? (3 points)

|  |
| --- |
| Answer: |

* 1. Describe the patterns in the graph. What does it say about the housing market in Providence, RI over time? In recent years? What additional data would you need to make claims about what is changing this price? (4 points)

|  |
| --- |
| Answer: |

1. **TEXT DATA (40 points total)**
2. Download US Economic News Dataset from Kaggle.com: Sign up for a free account with Kaggle.com. This website hosts data science competitions and often has cool datasets available for download. We will be using the US Economic News Dataset at <https://www.kaggle.com/heeraldedhia/us-economic-news-articles>. Download the CSV file from the website by clicking “Download.”
3. Load the data into Jupyter/Spyder/Python: Sometimes you may be working with a large dataset and it is therefore important to understand how to load a subset of the data at a time. The US Economic News dataset has 8,000 observations.
   1. Run the code below and explain in words each of the lines of code in the box to the right: (5 points)

|  |  |
| --- | --- |
| Code | Explanation: |
| ﻿import os  import csv  folder\_path = ‘TYPE FILE PATH TO WHERE YOU HAVE THE DATA’  fileReader = open(os.path.join(folder\_path,"US-Economic-News.csv"), "r", encoding="unicode\_escape")  csvReader = csv.reader(fileReader)  fileWriter = open(os.path.join(folder\_path,"Subset\_US\_Economic\_News.csv"), "w", encoding="unicode\_escape", newline='')  csvWriter = csv.writer(fileWriter)  acHeader = next(csvReader)  csvWriter.writerow(acHeader)    for index, acRow in enumerate(csvReader):  if index< 800:  csvWriter.writerow(acRow)  fileReader.close()  fileWriter.close()  df\_news = pd.read\_csv(os.path.join(folder\_path,"Subset\_US\_Economic\_News.csv"), encoding = 'unicode\_escape')  ﻿df\_news['date'] = pd.to\_datetime(df\_news['date']) |  |

* 1. What code would you write to keep only the ‘date’, ‘headline’, and ‘text’ columns in the dataframe? Run that code. (2 points)

|  |
| --- |
| Answer: |

1. This dataframe is full of text data about US Economic News. When we try to extract information from text, formatting of words and string in code is very important.
   1. Count the number of headlines that have ‘US’ in them. Hint: loop over df\_news[‘headlines’]. (3 points)

|  |
| --- |
| Answer: |

* 1. Count the number of headlines that have ‘us’ in them. (3 points)

|  |
| --- |
| Answer: |

* 1. Why are these counts different? Hint: tell python to check if ‘us’ is in the string ‘trust’. Then tell python to check if ‘ us ’ is in the string ‘trust’. (3 points)

|  |
| --- |
| Answer: |

1. In text analysis, we will need to perform a few tasks to clean the data to prepare it for consistent analysis. Run the code and explain what each line does: (5 points)

|  |  |
| --- | --- |
| Code | Explanation |
| ﻿  import nltk  from nltk.corpus import stopwords  nltk.download('stopwords')  stops = set(stopwords.words('english'))  table\_punctuation = str.maketrans('', '', '!"#$%&\'()\*+,-./:;<=>?@[\\]^\_`{|}~')  token\_list=[]  for i, row in enumerate(df\_news['text']):  text = row.translate(table\_punctuation)  tokens = [word.lower() for word in nltk.tokenize.word\_tokenize(text) if word.lower() not in stops]  token\_list.append(tokens)  df\_news['tokens'] = token\_list  ﻿monetary\_policy\_wordlist= ['monetary', 'fed ',  'federal reserve', 'Federal Reserve', 'Monetary']  tally=0  monetary\_text=[]  for row in df\_news['text']:  mon=0  if any(keyword in row for keyword in monetary\_policy\_wordlist):  tally+=1  mon=1  monetary\_text.append(mon)  print(tally)  df\_news['monetary\_flag'] = monetary\_text  df\_monetarynews = df\_news[df\_news['monetary\_flag']==1]  df\_nonmonetarynews=df\_news[df\_news['monetary\_flag']!=1] |  |

1. Compare and contrast the news articles about monetary policy in the US and those about non-monetary-policy economics in the US.
   1. This code calculates the top 30 most common words in df\_news. Adapt it for the following subquestions:

|  |
| --- |
| ﻿from collections import Counter  top\_N=30  words = [word for tokenlist in df\_news['tokens'].tolist() for word in tokenlist]  topwords = pd.DataFrame(Counter(words).most\_common(top\_N),  columns=['Word', 'Count']).set\_index('Word')  print(topwords) |

* + 1. What are the 15 most common words from df\_monetarynews? (3 points)

|  |
| --- |
| Answer: |

* + 1. What are the 15 most common words from df\_nonmonetarynews? (3 points)

|  |
| --- |
| Answer: |

* + 1. What differences do you notice? (1 points)

|  |
| --- |
| Answer: |

* 1. Building on the code from part a, you will visualize the word use in the different types of articles using a word cloud. Below is the code for making the word cloud for the df\_news dataframe. You must adapt it to the other dataframes in part i-iii:

|  |
| --- |
| ﻿﻿from wordcloud import WordCloud  ﻿allwords=' '.join(words)  word\_cloud=WordCloud(collocations=False, background\_color='white').generate(allwords)  plt.imshow(word\_cloud, interpolation='bilinear')  plt.axis("off")  plt.title('Word Cloud for US Economics Articles')  plt.show() |

* + 1. What is the word cloud for df\_monetarynews? (3 points)

|  |
| --- |
| Answer: |

* + 1. What is the word cloud for df\_nonmonetarynews? (3 points)

|  |
| --- |
| Answer: |

* + 1. What differences do you notice? Do these differences seem consistent with your list of top 15 most common words? (1 points)

|  |
| --- |
| Answer: |

1. Monetary Uncertainty in the News: Loughran and McDonald (2011) have created a commonly used bank of word-sentiment lists. One list is a list of “uncertainty words” You can find this dataset in the Github HW1 Repository. The following is code to make a Monetary Uncertain Score from df\_monetarynews and to plot the figure over time. However, there are three things wrong with in the code. Identify the typos, run the correct code, and insert the graph below. HINT: Run the code line by line and manually view the objects that were created and/or the error codes that appear. (5 points)

|  |
| --- |
| Code: |
| ﻿  folder\_path = ‘TYPE FILE PATH TO WHERE YOU HAVE THE DATA’  #Word Lists  uncertainty\_wordlist\_LM = pd.read\_csv(os.path.join(folder\_path,"LM\_Uncertainty.csv"), encoding = 'utf-8')  uncertainty\_wordlist\_LM = uncertainty\_wordlist\_LM['uncertain words'].tolist()  # Text Uncertainty Score for Each Article  uncertainty\_score=[]  for row in df\_monetarynews['tokens']:  u\_tally=0  for word in uncertainty\_wordlist\_LM:  if word in row:  u\_tally+=1  uncertainty\_score.append(u\_tally)    df\_monetarynews['text\_uncertainty\_score'] = uncertainty\_score    # Plot Yearly Mean Monetary Policy Uncertainty Over Time  #Take mean over years  df\_monetarynews['year'] = df\_monetarynews['date'].dt.year.astype(str)  df\_monetarynews\_yearly = df\_monetarynews.groupby(by=['year']).average()  df\_monetarynews\_yearly['year'] = pd.to\_datetime(df\_monetarynews\_yearly.index)  #Plot Time Series  plt.plot(df\_monetarynews\_yearly['year'],df\_monetarynews['text\_uncertainty\_score'])  ﻿plt.xlabel('Years')  plt.ylabel('Mean Uncertainty Score, Yearly')  plt.title('Uncertainty in US Monetary Policy News Articles')  plt.show() |
| Graph: |
| <insert graph here> |