Final Project: Data Analysis with Python

Outcomes

This project intends to bring together many of the skills we have (and will) talk about in the course. You will get a taste of modern Data Science using Python. Although the analysis I expect from you is not meant to be novel, we will use a modern technology stack, state of the art methodology, and practice creating and presenting analysis.

Methods and Technology Stack

We will be using **Jupyter and Git/GitHub** for working on the project.

Github link: https://classroom.github.com/a/7VROyLE-

We use Jupyter because it allows us to continue to develop code and visualize results without needing to rerun lengthy file processing, visualizations, and data analysis. We will be making scripts, not programs because the results are initially meant to be human-readable. There is no auto-grading or testing of your code. It will be evaluated based on the results you generate and the code itself.

You can work with a partner for this assignment, but if you decide to try to find your own data set, you can work in groups of 3 max. We will use GitHub to "collaborate" with your partner(s). Periodically, I will look at your GitHub to make sure you are on track with the project and provide some feedback.

You must demonstrate skill in Git by using either the Git CLI or a GUI to pull changes from your teammate(s), merge those changes with your own, and push incremental commits. You will lose points if you use the GitHub website to make code changes, and a component of the grade is from the Git history.

Assignment

For the project, you will solve a problem using data by:

- Progammatically download a data-set for analysis. The data is available here: GitHub
- Load the data-set into Python using a data structure (Data Frame).
- Understand and interpret the data-set.
- Visualize aspects of the data-set.
- Use Machine Learning techniques to solve the problem and perform the analysis.
- Customize your project through extension tasks.

Options

For the final project, you must choose 1 of 3 options. The first two are more structured and meant for the majority of students. The last option is an open project meant for those of you with more programming experience or a passion for a certain data-oriented problem (it will be a bit more work since you need to find your own data set).

The options are ordered by the expected difficulty:

- Predicting Housing Prices (Linear Regression)
- Detecting Fake News (Support Vector Machine)
- Open-Ended

Some resources to help you get started with finding a dataset:
https://data.boston.gov/
https://sports-statistics.com/sports-data/sports-data-sets-for-data-modeling-visualization-predictions-machine-learning/
https://www.kaggle.com/datasets

I will spend roughly 2 hours (split up) on various parts of the project. This is meant to be a challenging project with no provided starter code. I expect you to use Google, Pandas, and SKLearn documentation to solidify your understanding of the tools we are using in this project. I will introduce many of them, but will not spend much time explaining why they work and their inner-workings.

What I Will Show In Class:

- How to programmatically download a file
- Reading a csv with Pandas
- Looking at a dataframe
- Filtering columns
- Making a scatterplot
- Splitting data into Train and Test Data
- SKLearn Linear Regression with dummy data
- SKLearn SVM with dummy data
- Suggested Columns

Criteria	Coding Points	Write-Up Points
Uses Pandas + sklearn and explanation of any other tools used	5	1
Meaningful graphs/visuals with analysis	10	10
Working ML elements and robust explanations	15	10
Train/Test splitting and how you used it and why it is important	5	2
At least 2 extension tasks with analysis (3 for groups of 3)	10	15
Introduction, conclusions, and takeaways	0	12
Documentation (in README or writeup) on how to reproduce all results	5	0
A separate script to download a data file (optional extra credit)	2	0

Up to 5 points can be lost for bad Git/GitHub practices.

Up to 10 points of extra credit points may be awarded for students who submit remarkable work described along the following 3 parameters:

- Program quality (Professionally written code)
- Program functionality (Additional or creative extensions)
- Written work quality (Detailed report)

Discussion Writeup

You are expected to include a discussion of no more than 5 pages that explains the purpose of your work.

The discussion should include but is not limited to:

- A description of the problem you tried to solve
- The files you included and what they do
- Methods and Techniques you used along with "brief" explanations of how they should work
- What extensions you implemented and how they make your project unique and valuable
- Conclusions of the project, including what you've learnt

Submission Deliverables

2 points are deducted for failing to meet a deliverable on time.

Each deliverable must be accompanied by a quick summary of changes you made in the readme. This commit must alternate between members of the group to demonstrate knowledge of collaboration with Git. These are minimum deliverables and should be exceeded for more open ended work.

July 8th: GitHub Project Exists with README.md including your project topic. GitHub Classroom link

- If you are choosing the open-ended project, you must include a brief explanation and justification of your data-set and the problem you intend to solve/explore. Feedback provided after this deliverable for open ended projects.

July 22nd: Code that loads dataset into dataframe and some interesting exploration. Feedback provided after this deliverable.

August 5th (optional): Interesting exploration of data with python generated visuals. Some machine learning elements implemented.

August 9th: Final completed project with:

- Script to generate analysis
- README on how to reproduce results
- PDF write-up

Option 1: House Price Predictor

Overarching goal: Create a service to predict housing price

House Price Prediction

- Get and read a large data set
- Get an understanding of what is going on
- Process and wrangle the data
- Build a simple regression model using SKLearn
- Improve the model and do data analysis

Things to Research

- Making a network request for data
- Pandas, using a dataframe, plotting
- SKLearn and linear regression model

Extension Ideas

- Implement a linear regression class from scratch
- Visualize and evaluate the data in complex and creative new ways
- Use quantitative methods to intelligently pick features
- Compare different regression models
- Create a simple UI that can let people enter information and get an estimate of their house price.
- Visually and descriptively compare your results to other data sets

Things I'll show you in class

- Reading a csv with Pandas
- Looking at data (Seeing columns, filtering, making a scatter plot, etc.)
- A couple of suggested columns
- Splitting data into Train and Test Data
- SKLearn simple model with dummy data

Option 2: Fake News Detector

Overarching goal: Create a fake news detection algorithm to help fight the spread of misinformation before the 2020 election

Detect Fake News

- Get and read large data set
- Process and wrangle the data
- Gain an understanding of what is going on
- Vectorize content
- Use a Support Vector Machine to detect fake news

Things to Research

- Making a network request for data
- Pandas, using a dataframe, plotting
- SKLearn and classifier model

Extension Suggestions

- Compare other classifer models
- Use the article title
- Create a simple UI to upload and test articles
- Using your classifier, analyze and compare several news sites

Things I'll show you in class:

- Reading a csv with Pandas
- Looking at data (Seeing columns, filtering, making a scatter plot, etc.)
- A couple of suggested columns
- Splitting data into Train and Test Data
- SKLearn simple model with dummy data

Option 3: Unguided Final Project

Things You Must Do

- Find a large (>1000 rows) data-set online
- Programmatically download the data-set
- Load the data into a Pandas Dataframe
- If needed, filter the data to remove unwanted rows and columns
- Use Pandas or Matplotlib to create at least 3 interesting visualizations of meaningful data
- Use Regression or Machine Learning through SKLearn to perform meaningful analysis on the data (A predictive element)
- Extend your program to do something advanced
  - Allow user input to interact with your model (predict results)
  - Compare multiple (at least 3) different methods for doing your data analysis
  - Answer real-world questions that pertain to your data (extend the analysis)