Introduction to Python

Fondren Library
Research Data Services

Creating Software with Python

Final Project Tip 1: Start with a Clear Plan

Define the scope and objectives of your project. Ensure you understand what you want to achieve and the steps needed to get there.

Jakub - python program would input user info and narrow down nyc real estate searches based on housing needs using future RE predictions

Michelle - centenarian would like to see locations individuals are living over 100yrs of age and what is the diet and lifestyle of individuals

Cristina: Topic: Daily Wellness and Creativity Enhancer, for morning people and generate personalized workout routines and inspire users to write

Carolina: Program that compiles different activities and programs for differently abled people, where their guardians can find information, and find resources to pay for them. I need to gather the data, and design the steps needed to get to a useful program.

Mandi- Automating Coordination of Disability Services

Ikuko - Automating visualization process from spreadsheet data

Monica - chord finder, input a note of the scale and generate chords, and sample progressions to help people write and read music

JC - Topic: Impact-Financing-Placement System for Agriculture in Peru

Anna - Collect and analyze social media data to understand public sentiment on mental health issues

Jonathan - Book rec using weighted tags

Final Project Tip 2: Break Down the Task

Divide your project into manageable tasks. Create a timeline to keep track of progress and deadlines.

Jakub - pace yourself by doing a couple of hours a day instead of crunching everything at the end. Step1: create user input program. Step2: link basic program with non-RE related data to make more complex

Mandi- Researching databases; deciding what data need; Breaking down eligibility criteria per service

Cristina: Looking for data about exercises with weights, time and rounds, elaAnna - Collect and analyze social media data to understand public sentiment on mental health issues

Anna - Use APIs provided by social media platforms to collect data. Remove duplicates and filter irrelevant data. Use python libraries for basic analysis. Identify trends from the data and use visualization.

Monica - simplify dataset, determine features, code

Ikuko
- Understand data structure
- Understand Google Applications
- Finalize with coding

Jonathan
-Acquire viable dataset/s
-Determine features
-Coding

Carolina:
Think what data I need,
Collection of data
Understand what questions de data can solve
Design a program that can accomplish the goal
Test
Make it user friendly

orate some prompts to inspire and users can engage with journaling. Avoid procrastination, check for some help to make time to develop the idea.

Michelle: Work with the data to extract from the demographic the information needed to build out my program

IC

1. Validated the source
2. Collected the data (xlsx)
3. Stored the data (Local drive, OneDrive - Sharepoint, GitHub)
4. Develop the relational model
5. Translating the data into Python and saved in jupyter format
6. Identifying programming challenges
7. Visualizing my thought process
8. Exploring data with ER Diagram
9. Analyzing the data with python!

Final Project Tip 3: Choose the Right Dataset

Ensure your dataset is relevant and clean. Consider the size, quality, and source of the data. Use reputable sources and verify the data's accuracy.

Jakub - using Zillow's historical RE data and the "Zestimate" by focusing on different categories. Cleaning said categories, making predictions and then combining said predictions with outside sources such as crime, 311 complaints to guide the user in the expected individual housing needs

Carolina:
Datasets form NYC official websites that are relevant to what I need to do.

Cristina: Using Kaggel and check for comments, reviews to confirm is a good source

Michelle: not sure how to answer this one, I have started with making sure the data set collected is from a reliable 'census' site

Anna: Will probably use Twitter API, but not entirely sure yet.

Mandi: Look through data sets from disabilitystatistics.org, data.world, data.gov, worldpolicycenter.org

Jonathan
-Option 1: web scraping for data
-Option 2: public book datasets available

Ikuko
- Dataset: Raw digital ads campaign data in spreadsheet
- Size and quality is good since it will be of a single campaign
- The source will come from the equivalent ads platform

JC

1. Validated the source
2. Wrangled the data (Excel, Notepad, MS Data Wrangler for VS Code)
3. Included id columns to facilitate relations
4. Created clean final csv files with wrangled relational data

Final Project Tip 4: Utilize Libraries and Frameworks to your Advantage

Leverage Python libraries like Pandas for data manipulation, NumPy for numerical computations, Matplotlib and Seaborn for visualization, and Scikit-learn for machine learning tasks.

Jakub - pandas: add missing values, join categories. Numpy: take averages. matplotlib/seaborn: graphs and plots for final project presentation. Scikit-learn: learn the user?

Michelle: pandas to clean and manipulate my data set

Cristina: Pandas for all manipulation, some machine learning, but I am not sure yet.

Mandi-Pandas to clean, explore and manipulate data; Still trying to figure out this connection to the end result

JC

1. Transformed my final csv files into jupyter dataframes
2. Installed a python package for VS Code for impact analysis

Ikuko
Still in the process of deciding, but potentially Pandas for cleaning the data.
For visualization, trying to utilize Google Slides API using JSON.

Jonathan
-Pandas, NumPy, Scikit-learn for data manipulation, cleaning and usage
-Need a library for web

Anna: Will probably use pandas and try other libraries.

Carolina:
Possibly pandas, and while I plan better what I want to do I may use something else.

Final Project Tip 5: Perform Exploratory Data Analysis (EDA)

Spend time understanding your data. Use EDA to identify patterns, anomalies, and relationships in the data. Visualize data using plots and charts.

Jakub - use previous mentioned libraries to get a quick snapshot glimpse of the data at hand

Cristina: To understand my data I could use machine learning

Mandi: Spend a few hours exploring data sets and uncovering possible hidden relationships; document relations and reflect if this would be important to present in program

Michelle:

Anna: I'm not sure

JC

- Used Lucid charts for ER Diagramming the relation of the data.
- Identified the categorical axis' for data relationships
- Integrated id columns to consolidate the data relationships across the dataset

Jonathan
-Patterns- Same names, similar tags, different ratings
-Anomalies-Missing tags/data

Ikuko
Use EDA to understand if there are any NA data and try this on 2-3 multiple sample data.
Visualizing will be a necessary step for the output as well.

Final Project Tip 6: Feature Engineering

Transform raw data into meaningful features that improve the performance of your model. Consider normalization, encoding categorical variables, and creating new features from existing data.

Carolina: Realize how the data can be analyzed and create different features that can improve the answers that I can provide the users of my program.

Jakub - use existing data to create new tables/categories. Ex: future prediction RE data

Anna:

Mandi-creating categories and tailoring services to each feature/combo you would like to present

Cristina:

- Type of exercise
- Equipment

JC

1. Based in the clean relational dataset:
   a. Breakdown the dataset into categorical variables representing the indicators I am working with
   b. Labeling consistently across the dataset
   c. Encoding

Ikuko
Encode campaign types (categorical variables) for future analysis to understand the tendency (Performance by audience over time)

Jonathan
-grouping tags
-origin country/language
-publishing date
-ratings/popularity
-encode categorical data

Final Project Tip 7: Model Selection and Tuning

Choose appropriate models for your problem. Experiment with different algorithms and use techniques like GridSearchCV to tune hyperparameters for optimal performance.

Jakub - by taking user data begin to get to know them, so for example if they turn off/on the program and return, you already "know" them/their info

Cristina I will choose the best model doing some test to my data

Mandi- Not sure what model will work best yet; Choose model and test data sets for outcomes/relationships/results; looking for flaws

JC
Need to work on this

Carolina:I am not sure if I will need it.

Ikuko
Deciding what and how to apply the model in my project

Jonathan - optimize model for recs with limitations

Anna: I would like to experiment with a model for my project

Final Project Tip 8: Validate Your Model

Use cross-validation techniques to ensure your model generalizes well to unseen data. Evaluate performance using metrics like accuracy, precision, recall, and F1 score.

Jakub - be open to user recommendations to expand inputs. Create wants/musts category. Make assumptions on high income users: rich people like expensive things -> assume rich people want expensive RE in set range.

Carolina:I am not sure that my project is so complex to need this.

Michelle: I can cross-check my work from the dataset used from the census with a documentary I saw in NetFlix to how I want to model out my work

Mandi-testing data for accuracy

Cristina Make some testings and see what is the best option

Ikuko
With the visualization part of my project cross-validation may not be needed.
For the future feature it would be interesting to evaluate accuracy on the overall analysis to understand campaign performance trend by audience.

JC
Need to work on this

Jonathan
- Test how system works with adding/updating books/limitations

Final Project Tip 9: Document Your Work

Keep thorough documentation of your code, processes, and findings. This will make it easier to present your project and for others to understand your work.

Jakub - combine short code spurts with visual representation. Break code down into sections.

Mandi

- Timeline
- Activity Log
- Summary: Blocks of Time visually representing how work was divided up, processes, time and work notes in each category

Carolina:

Cristina

Ikuko
Create a project document to understand the goal of the project, target users, impact and timelines

JC
Need to work on this

Jonathan
- code documentation
- activity log

Final Project Tip 10: Seek Feedback

Share your progress with peers or mentors and seek constructive feedback. Incorporate suggestions to improve your project.

Jakub - weekly check-ins with class peers for feedback

Carolina: Try to keep up the phase and use this space to be able to comment my project with my peers and mentor so they can give me ideas and feedback.

Mandi:
Check in with professor and peers for feedback, blind spots, commentary, ideas, great things and possible concerns

Ikuko
Progress: Still in progress of adding more visualization to the slide I am trying to output
Share this with peers at work for more FB

Jonathan
- ask for feedback

JC
I have been sharing with my mentor permanently.

Technique 1: Data Cleaning

Example: Removing missing values and duplicates from a dataset.

```
import pandas as pd

# Load dataset
df = pd.read_csv('data.csv')

# Remove rows with missing values
df_clean = df.dropna()

# Remove duplicate rows
df_clean = df_clean.drop_duplicates()

print(df_clean.head())
```

Technique 2: Data Visualization

Example: Removing missing values and duplicates from a dataset.

```
import matplotlib.pyplot as plt


# Load dataset
df = pd.read_csv('data.csv')


# Create a histogram of the 'age' column
plt.hist(df['age'], bins=20)
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Age Distribution')
plt.show()
```

Technique 3: Feature Scaling

Example: Normalizing data using MinMaxScaler.

```
from sklearn.preprocessing import MinMaxScaler # Sample
data data = [[-1, 2], [-0.5, 6], [0, 10], [1, 18]] #
Normalize data scaler = MinMaxScaler() scaled_data =
scaler.fit_transform(data) print(scaled_data)
```

Technique 4: Encoding Categorical Data

Example: One-hot encoding categorical variables.

```
import pandas as pd

# Sample data
df = pd.DataFrame({'color': ['red', 'blue', 'green',
'blue', 'red']})

# One-hot encoding
df_encoded = pd.get_dummies(df, columns=['color'])

print(df_encoded)
```

Technique 5: Dimensionality Reduction

Example: Applying PCA to reduce dimensionality.

```
from sklearn.decomposition import PCA
import numpy as np

# Sample data
data = np.array([[2.5, 2.4], [0.5, 0.7], [2.2, 2.9], [1.9, 2.2], [3.1, 3.0],
[2.3, 2.7], [2, 1.6], [1, 1.1], [1.5, 1.6], [1.1, 0.9]])

# Apply PCA
pca = PCA(n_components=1)
reduced_data = pca.fit_transform(data)

print(reduced_data)
```

Technique 6: Regression Analysis

Example: Performing linear regression.

```
from sklearn.linear_model import LinearRegression


# Sample data
X = [[1], [2], [3], [4], [5]]
y = [1, 3, 5, 7, 9]


# Model training
model = LinearRegression()
model.fit(X, y)


# Predicting
prediction = model.predict([[6]])
print(f"Prediction for input 6: {prediction[0]}")
```

Technique 7: Classification

Example: Using RandomForestClassifier.

```
from sklearn.ensemble import RandomForestClassifier


# Sample data
X = [[1, 2], [3, 4], [5, 6], [7, 8]]
y = [0, 1, 0, 1]


# Model training
model = RandomForestClassifier()
model.fit(X, y)


# Predicting
prediction = model.predict([[4, 5]])
print(f"Prediction for input [4, 5]: {prediction[0]}")
```

Technique 8: Clustering:

Example: Using K-means clustering.

```
from sklearn.cluster import KMeans
import numpy as np


# Sample data
X = np.array([[1, 2], [1, 4], [1, 0], [4, 2], [4, 4], [4, 0]])


# Model training
kmeans = KMeans(n_clusters=2)
kmeans.fit(X)


# Predicting clusters
clusters = kmeans.predict(X)
print(f"Clusters: {clusters}")
```

Technique 9: Time Series Analysis

Example: Plotting a time series.

```
import pandas as pd
import matplotlib.pyplot as plt


# Sample data
dates = pd.date_range('20230101', periods=6)
df = pd.DataFrame({'values': [1, 3, 5, 2, 4, 6]}, index=dates)


# Plotting time series
df.plot()
plt.title('Time Series Plot')
plt.show()
```

Technique 10: Natural Language Processing (NLP)

Example: Tokenizing text data using NLTK.

```
import nltk
from nltk.tokenize import word_tokenize


# Sample text
text = "Natural language processing with Python is fun."


# Tokenizing text
tokens = word_tokenize(text)


print(tokens)
```

Activity: Final Project Discussion