TUTORIAL NOTES
FRANCESCA BORG

WORK ON TUTORIALS

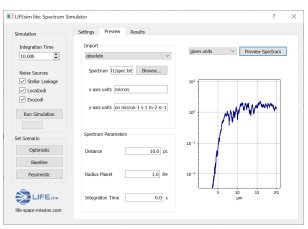
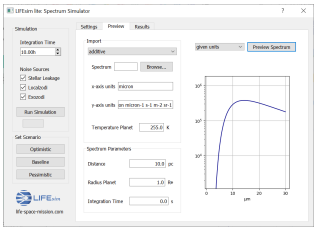
TUTORIAL 1: LIFESim

Jupyter notebook: *Simulating_exoplanet_search.ipynb*
Resources: [1] LIFEsim demo tutorial
[2] GitHub introductory tutorial
[3] Getting started with GitHub

Date: 16 Mar 2024, **Hours**: 2 PM - 8 PM
Goals: set up GitHub + organise account
Work progression:
- Set up GitHub account
- Started following GitHub tutorials [2], [3]
- Created repository for coursework
- Downloaded Git bash

Date: 20 Mar 2024, **Hours**: 7 PM - 8 PM
Goals: set up LIFEsim for tutorial
Work progression:
- Installed miniconda + Jupyter notebook
- pip installed LIFEsim package

Date: 21 Mar 2024, **Hours**: 10 AM - 12 PM (Tutorial)
Goals: simulate catalogue of exoplanets, simulate spectrum of exoplanet
Work progression: following tutorial [1]
- used LIFEsim GUI to simulate spectrum of a single exoplanet with a given spectrum with the following settings to obtain the following graphs:

- Run the simulation in three different scenarios (pessimistic/baseline/optimistic)
- **Innovations**: experimenting with different parameters
- **Problems encountered:** tutorial [1] used commands from older versions of LIFEsim which have since been updated. **Problems resolved:** updated versions of the commands found + used. **Relevant part of the code** is marked on the nb.

Date: 27 Mar 2024, **Hours**: 11 AM - 2 PM, 5 PM - 8 PM
Goals: following tutorial [1], upload nb to GitHub
Work progression:
- Continued tutorial [1]
- **Problems encountered:** 0 exoplanets detected output (code not working as expected).
- Compute the correlation matrix: correlated features can be excluded from the Neural Network
- Cloning Git repository to uploading Jupyter nb 'Simulating_the_Search_Phase' to GitHub

Date: 4 Apr 2024, **Hours**: 10 AM - 12 PM
Goals: compute correlation matrix
Work progression:
- **Problems resolved:** computing correlation matrix
- Removing non-numerical data

- Removing columns which do not matter in correlation e.g. object id (‘id’, ‘name_s’), position of star/planet (‘ra’, ‘dec’, ‘lon’, ‘lat’), ‘baseline’ and zero columns ‘ecc_p’

Date: 01 May 2024, **Hours**: 4 PM - 7 PM
Goals: finish off tutorial 1 + upload final notebook to GitHub
Work progression:

- **Problems encountered:** attempting to print names of highly anti/correlated variables

TUTORIAL 2: Naive Bayes

Jupyter notebook: *Naive_Bayes.ipynb*
Resources: [1] Naive Bayes colab nb

Date: 28 Mar 2024, **Hours**: 10 AM - 12 PM
Date: 02 April 2024, **Hours:** 12 PM - 2 PM
Date: 07 April 2024, **Hours:** 4 PM - 7:30 PM
Date: 08 April 2024, **Hours:** 10 AM - 1 PM
Goals: implementing Naive Bayes classifier: follow colab notebook example, make own case of Naive Bayes by using some exoplanet data
Work progression:

- Read through colab notebook + run examples
- Naive bayes classification of LIFEsim data to predict habitability
- Naive bayes classification of exoplanetary detection dataset from NASA to predict detectability by transit method
- **Problems encountered:** confusion matrix showed bad results for LIFEsim data
- **Problem resolved:** tried Naive Bayes classification with a different dataset to see whether it performs better
- Added theoretical background on supervised vs. unsupervised learning algorithms

TUTORIAL 3: ML techniques

Jupyter notebook: *exoplanet-exploration-using-ml.ipynb*
Resources: [1] Exoplanet-exploration
[2] Exoplanet time series data

Date: 4 Apr 2024, **Hours**: 10 AM - 12 PM
Date: 8 May 2024, **Hours:** 2 PM - 6:30 PM
Date: 9 May 2024, **Hours:** 7 AM - 8 AM
Goals: Some of Machine learning techniques applied on classification of exoplanets using programs by Data analyst Nagesh Singh Chauhan.

- create a new repository on your github account which name will be EXOPLANETS_ML https://docs.github.com/en/get-started/quickstart/create-a-repo
- clone Nagesh Singh Chauhan exoplanet repo https://github.com/nageshsinghc4/Exoplanet-exploration into your github EXOPLANETS_ML Repo using instructions in GITHUB_CLONE.pdf
- when cloning is done, go to *exoplanet-exploration-using-ml.ipynb* and be sure to run through the notebook; the aim is that you understand each line of the code, make printing of intermediate results, and if unknown command is encountered than to go to the online sources to understand its meaning
- could you make some innovations in the ipynb such as new functions, plots, import other data, etc.

Work progression:

- Created new repository and cloned Exoplanet-exploration
- **Problems encountered:** older versions of sklearn, sns were used in writing this code. **Problems resolved:** some commands had to be modified to run
- **Problems resolved:** oversampling to fix imbalance in habitability of our data

TUTORIAL 3-1: Support Vector Machines

Jupyter notebook: *Support_Vector_Machines.ipynb*
Resources: [1] In Depth: Support Vector Machines

Date: 9 May 2024, **Hours**: 10 AM - 12 PM, 3 PM - 7 PM
Goals: Go through colab notebook, innovation
Work progression:

- Read through notebook
- Innovation: using exoplanet data from NASA with a SVM to sort between different exoplanet detection methods
- Experiment with effect of parameter C on the margin of the Support Vector Classifier (SVC) boundary
- Uploaded NB to github

TUTORIAL 4: Simple Artificial NN

Jupyter notebook: *Exoplanet_ANN.ipynb*
Resources: [1] nageshsinghc4/Exoplanet-exploration
[2] Build an ANN from scratch (part 2).html
[3] ANN-from-scratch-python (repo)

Date: 9 May 2024, **Hours**: 7 PM - 8 PM
Date: 10 May 2024, **Hours:** 8 AM - 10:30 AM
Goals: Simple Artificial neural network technique applied on classification of exoplanets using programs by Data analyst Nagesh Singh Chauhan
Tasks:

- Use the same repository as Tutorial 3 (EXOPLANETS_ML)
- Go to *exoplanet_ANN.py* and try to make ipynb of it
- Take a look at article linked in repo
- Go through notebook + understand code + print intermediate results
- Make some innovations in .ipynb such as new functions, plots, import other data, etc.

Work progression:

- Created *Exoplanet_ANN.ipynb*
- Went through notebook
- Explored new functions and plots re: building an ANN from scratch from external source [2]
- Uploaded complete notebook to GitHub

TUTORIAL: Feature PCA

Jupyter notebook: *Principal_Component_Analysis.ipynb*
Resources: [1] 05.09-Principal-Component-Analysis.ipynb

Date: 10 May 2024, **Hours**: 7 PM - 8 PM
Date: 22 May 2024, **Hours:** 3 PM - 9 PM
Goals: Feature selection and dimensionality reduction. Introduce Principal component analysis for feature selection and data dimensionality reduction.
Tasks:

- pass through the notebook https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/05.09-Principal-Component-Analysis.ipynb
- Make innovations in the notebook

Work progression:

- Created notebook *Principal_Component_Analysis.ipynb*
- Read through tutorial
- Innovation: applied PCA to exoplanetary data (NASA catalog)
- **Problem encountered:** draw_vector method from tutorial notebook not printing on the output plot
- **Problem resolved:** using wrong dataset (non-logged X)
- Finalised + uploaded notebook to GitHub

TUTORIAL 5: Application of Transformer Model on Classification of Habitable Exoplanets from LIFE Sim

Jupyter notebook: Attention_mechanisms_in_transformer_models.ipynb
Resources: [1] Shared google drive
[2] RNNs and Transformers for Sequence-to-Sequence Modeling.ppt

Date: 23 May 2024, **Hours**: 10 AM - 12 PM, 6 PM - 9 PM
Date: 28 May 2024, **Hours**: 9 AM - 1 PM, 6 PM - 9 PM
Goal/s: Apply a transformer model to classify habitable exoplanets from LIFESim. Tasks:

- Understand the Transformer Architecture:
  - Study the core components of the Transformer model (e.g., self-attention mechanism, positional encoding, multi-head attention).
  - Learn how these components contribute to the model's performance in various tasks.
- Apply the Transformer Model:
  - Use a pre-existing dataset from the LIFE Simulation to classify habitable exoplanets.
  - Implement the model in a Jupyter notebook, ensuring you can run it on your own computer.
- Set up the environment
  - Ensure you have Python and necessary libraries installed (e.g., PyTorch).
  - Install any additional packages required for working with the Transformer model.
- Understand data preparation
  - Loading the LIFE Simulation dataset.
  - Preprocessing the data, which might include normalization, handling missing values, and splitting the data into training and testing sets.
- Model implementation:
  - Understand the Transformer model architecture and implementation of the model using a deep learning framework of PyTorch.
  - Training the model on the training data.
  - Evaluation of the model's performance on the testing data.
- Innovations: analysis and visualisation
  - Analyse the model's performance using appropriate metrics (e.g., accuracy, precision, recall).
  - Visualise the results to better understand the model's performance and the data distribution.

Work progression:

- Went through code and followed slides [2] to understand how each component contributes to the transformer model's performance. **Innovation**: adding theoretical context to the code, creating a jupyter notebook.
- **Problems encountered:** error in code. **Problems resolved:** added .values to select value from the dataframe
- How many heads does the transformer have? The transformer model has only 1 head, but this can be changed by varying *num_heads*