Project List

Project 1

Given a DNA sequence dataset with two possible outcome variables—coding (1) and noncoding (0)—your task is to encode this data into vectors using an appropriate method, such as the K-mers approach or one-hot encoding. Then, build a classification model using Long Short-Term Memory (LSTM).

Download the CSV file containing the DNA sequences (DNA_set.csv). Use 80% of the DNA_set for training and 20% for testing when building your classification model. Save your model and compute its performance using the provided test set (test_set.csv) and the validation set (validation_set.csv). Compute all the necessary performance metrics, including F1 score, accuracy (ACC), the area under the curve (AUC), Matthews correlation coefficient (MCC), and the confusion matrix for both the test set and the validation set.

Using the same dataset, apply a transformer approach, specifically, use a pre-trained DNA language model such as DNABERT (https://github.com/jerryji1993/DNABERT). Compare the results of the LSTM and transformer model. Discuss your observations on their performances.

Project 2

Given a set of protein sequences (protein.csv), your task is to employ Chaos Game Representation (CGR) for dimensionality reduction to transform these sequences into images. CGR is a technique that maps sequences to graphical representations, providing a way to visualize the structure and patterns within the sequences. (Chaos game representation and its applications in bioinformatics. Comput Struct Biotechnol J. 2021 Nov 10;19:6263-6271. doi: 10.1016/j.csbj.2021.11.008). Once you have the CGR images, use an autoencoder to convert these CGR images into a low-dimensional vector space. This involves training an autoencoder to learn efficient codings of input data.

The goal of training the autoencoder using your CGR images as input is to minimize the reconstruction error, ensuring that the encoded vectors can be used to accurately reconstruct the original CGR images through the decoder. Once the autoencoder is trained, use the encoder part of the network to convert all CGR images into their corresponding low-dimensional vectors. Save these vector representations in a database.

Compute the performance of your autoencoder model based on its ability to decode the original sequences. This can be measured using metrics such as Mean Squared Error (MSE) or Structural Similarity Index (SSIM) between the original CGR images and the reconstructed images. A lower error indicates a better-performing autoencoder. Additionally, evaluate the ability of the autoencoder to reconstruct the original protein sequences from the decoded CGR images. This step ensures that the entire process, from sequence to vector and back to sequence, is accurately captured.

Project 3 Baseline: MAE = 10 Best: MAE = 6

In this project you are tasked with developing a CDSS in form of a so called aging clock model, which predicts the age of a patient given medically relevant variables. Such a supervised machine learning model, given sufficient performance results, can be used to determine the discrepency between a patients chronological age and biological age (by looking at the difference of actual age and predicted age). This information is useful for medical practitioners, for example in regards to the choice of age sensitive therapies.

In case of this project the aging clock shall be trained on human gut microbiome data from the American Gut Project (the relevant data can be found on Ilias). The American Gut project was a large scale citizen science project collecting the DNA of the microbiome of the participants which was then converted into a table, that gives the information which microbe was found how many times in the gut of each participant. This data is then to be used for training of the aging clock.

Evaluate your models via typical metrics such as the mean absolute error and the root mean squared error. Ideally your CDSS outputs the difference between actual age of a patient and predicted age with an indictaion whether the difference is considered large enough for medical relevance.

Project 4

In this project the goal is to create a rudimentary, ontology based CDSS which can function as a drug interaction checker. Your task will mainly entail the implementation or creation of a drug interaction ontology and the development of a sensible rules system that identify and highlight potential drug interactions. For an existing example look into the Drug Ineraction Ontology (DIO; https://bioportal.bioontology.org/ontologies/DINTO). The DIO could even be integrated into yours to enhance coverage.

A recommended tool for ontology related work is Protégé (https://protege.stanford.edu). However which tools you use for this project is lastly up to you.

The endresult should optimally include a user interface that makes it possible to check the interactions of any number of drugs included in the ontology.

Project 5

In this project you are tasked with the creation of a fully functional fuzzy logic-based CDSS from the ground up. The goal is for the CDSS is to aid in the diagnosis of a variety of different diseases by input of the patients symptoms. To optimize the validity of the diagnosis the input data should be a pooled dataset from different sources that share the inclusion of at least some symptoms.

The data collection step is part of your task for this project. As medical data is often hard to come by, try to find matching data from sites such as https://www.kaggle.com, https://archive.ics.uci.edu or by searching for matching studies via sites like https://scholar.google.com and checking the availability of their data.

Lastly the finished project should optimally be an easy-to-use software that takes in a patients symptoms and outputs their most likely affliction.

Project 6

In this project you are tasked with the creation of a supervised learning-based CDSS that predicts heart disease risk and/or diagnosis. To that end your task is train and evaluate the performance of different supervised machine learning algorithms.

Then determine the best model via common statistical metrics such as accuracy and ROC-AUC, and potentially even combine the best performing models in an ensemble learning approach.

The repository of the UCI has a suitable dataset for this task (https://archive.ics.uci.edu/dataset/45/heart+disease), but the pooling of similar data potentially inreases the performance of the models, so consider looking for larger/more data.

Lastly the output of the should be presented to the user in a easy-to-use and easy-to-read fashion.

Project 7

In this project you are tasked with the creation of a CDSS for breast cancer diagnosis based on x-ray images. Multiple different image data-suitable classification models can be tested and potentially combined. Image-typical feature reduction methods such as deep Autoencoders could preceed your classification model, or the images can be fed directly into a classification model such as a Convolutional Neural Network. Finding the best performing approach is part of your task for this project.

Suitable datasets can be accessed via the Cancer Imaging Archive of the American National Institute of Health (https://www.cancerimagingarchive.net/collection/breast-cancer-screening-dbt). This particular dataset is quite large, should that pose a problem, other smaller datasets should also be available for the project's purpose.

Optimally the project includes a report at the end on which image-features were the most relevant for the cancer classification, via common explainable AI methods.

Project 8

In this project you are tasked with the implementation and evaluation of different clustering methods with the aim of creating a CDSS that takes electronic health records of patients as an input and outputs distinct clusters into which each patient can be grouped. Which unsupervised learning methods or algorithms in general are tested is up to you.

Evaluate which tested method performs best by using statistic metrics such as silhouette score and Davies-Bouldin index. Then implement an intuitive way to see which features in the data are shared/have similar expressions between clusters.

Similar to project 4 finding a suitable dataset or maybe even multiple datasets that can be combined is part of your task. Electronic health records are rarely public so the same sources as listed in project 4 could be searched (https://www.kaggle.com, https://archive.ics.uci.edu, https://scholar.google.com).