BTEC 423: Machine Learning (ML) Applications for Translational Bioinformatics

Course Designer:
Jeffrey Robinson, MS, PhD
Translational Life Science Technology Program
UMBC at Universities at Shady Grove
9636 Gudelsky Dr., Rockville, MD, 20850
Phone: 301-335-4851
Email: jrobin2@umbc.edu

BTEC 423 (4 credit): Machine Learning (ML) Applications for Translational Bioinformatics. This is a senior-level capstone course for the TLST Bioinformatics Track. This course provides a knowledge-base and practical experience in ML for bioinformatics-specific applications in the analysis of large clinical and –omics datasets. Students will learn a comprehensive 'roundup' of ML algorithms with practical coursework focusing on bioinformatics applications and clinical data analytics. Large open-source datasets will be utilized such as the Framingham Heart Study and Omics studies from NCBI GenBank database and The Cancer Genome Atlas. Practical coursework will utilize the Python programming language with scikit-learn on Jupyter Notebooks (or equivalent).

Prerequisites: BTEC330 (Software Applications), BTEC350/STAT350 (Biostatistics), BTEC395 (Bioinformatics), BTEC362 (Python programming).

Course Objectives:

1. Understand the theory and applications of common ML algorithms.
2. Develop ML predictive models using Python3 and Jupyter notebooks IDE (Integrated Development Environment).
3. Apply ML for specific translational applications by programming two use-cases:
    a. patient risk stratification from a clinical dataset.
    b. analysis of a transcriptomics dataset.

Textbook and course content:

1. Irizarry and Love. "Data Analysis for the Life Sciences."
2. Sullivan "Python Machine Learning: Illustrated Guide for Beginners & Intermediates"
3. "Deep Learning by Example on Biowulf" class series, INCLUDES PYTHON CODE. https://hpc.nih.gov/training/

Grading:

Exam 1: 25%
Exam 2: 25%
Term Project (Due Finals Week): 30%
Homework: 15%
Attendance and Participation (at least five question/comment inside or outside of class): 5%

Course Outline.

1. **Module 1: Machine Learning Algorithms and Python Programming with Jupyter Notebooks**
   Weeks 1-5.
   Text: "Python Machine Learning: Illustrated Guide for Beginners & Intermediates"
   1. Coding effectively in the developer's environment.

a. Python3 using Jupyter Notebooks.
b. High Performance Cloud computing with the NSF Jetstream platform.

2. Data Preprocessing, "Data Wrangling".
   a. Combining and subsetting data with data frames.
   b. Utilizing metadata.
3. The common algorithms of Machine Learning with brief examples.
   a. Linear Regression, Polynomial Regression
   b. Decision Trees, Random Forests, Support Vectors for Regression
   c. Classification: Naive Bayes, K-Nearest Neighbors, Decision Trees for Classification
   d. Dimensionality Reduction: Hierarchical Clustering, PCA, LDA.
4. Performance evaluation and automated optimization: Cross Validation and Grid Search

Exam 1 (25% point) ML problem sets. Students will receive a dataset, and a series of questions which they must apply specific ML algorithms to develop basic analyses.

2. **Module 2: Machine Learning applications – Clinical data.**
   Weeks 6-10
   Text: selections from "Data Analysis for the Life Sciences"
   1. Factoring for demographic variables in the dataset (age, gender, race, BMI).
   2. Applying ML regression modeling for generating diagnostic models based on clinical blood test results.
   3. Review of large clinical-translational datasets such as the Framingham Heart Study and genomics datasets.

   Exam 2 (25%): Exam will test student understanding of ML regression methods and their applications for analysis of clinical datasets.

3. **Module 3: Machine Learning applications – Transcriptome data.**
   Weeks 11-15.
   Exercises and code adapted from "Deep Learning by Example on Biowulf" class series.
   https://hpc.nih.gov/training/
   1. Dimensionality Reduction – PCA analysis of transcriptome data from The Cancer Genome Atlas.
   2. Deep Learning (Auto-encoder (unsupervised)) for Classification of tumor subtypes based on gene expression.

   Term Project (30%): Students will adapt the code provided from the NIH series, but using the Jupyter notebook IDE. Students will generate the code as an open-source application on the GitHub platform. Projects will be graded on the basis of the functionality of their code and application of proper coding practices as delineated in the assignment description.