תרגיל מסכם

הגשה עד: 16.8.2022

הנושא השנה הוא אימון מודלי סיווג על נתוני microarray אשר משמשים לזיהוי של מוטציות גנטיות ושינויים בין פרטים ואוכלוסיות (ראו https://en.wikipedia.org/wiki/DNA_microarray). נתוני microarray מאופיינים על ידי מספר גדול של features (= הגנים) ויחסית מעט דגימות לאימון (חולים/בריאים) ולכן סובלים מ-"קללת המימדיות" (ראו https://en.wikipedia.org/wiki/Curse_of_dimensionality). מטרת התרגיל היא לבחון שיטות להקטנת המימדיות של הנתונים על ידי feature selection.

חלק א': בחירת אלגוריתמים ותיאורו (10 נק')

- לקבלת רקע כללי על המשימה, קראו את המאמר הבא:

Hambali, Moshood A., Tinuke O. Oladele, and Kayode S. Adewole. "Microarray cancer feature selection: review, challenges and research directions." *International Journal of Cognitive Computing in Engineering* 1 (2020): 78-97.

- כל זוג צריך לבחור שני מאמרים מתוך הרשימה בקישור: https://docs.google.com/spreadsheets/d/1Ejc_kZN0UNL5hLNTaYlJvPY3bTFWnMXWgBv7nO4C7qs/edit?usp=sharing
- יש לבחור מאמר אחד מהקבוצה Part A(מאמרים קלאסיים) ומאמר שני מהקבוצה Part B (שיטות חדשות)
- בחירת האלגוריתמים תעשה לפי First Come First Serve. יש לרשום את שמות חברי הקבוצה ליד המאמר הנבחר. ברגע שקבוצה בחרה במאמר, קבוצה אחרת לא יכולה לבחור את אותו המאמר.
- יש לממש את האלגוריתם המתואר בכל מאמר. אם במאמר מתואר יותר מאלגוריתם חדש אחד יש לממש את האלגוריתם המפיק את הדיוק הרב ביותר.
- תארו את הפסאודו קוד של האלגוריתמים שנבחרו.
- שייכו את האלגוריתם לטקסונומיה המתוארת בתרשים 3 במאמר [Hambali et al. [2020
- תארו את היתרונות והחסרונות של האלגוריתמים שנבחרו.
- הדגימו לפחות שלוש איטרציות של האלגוריתמים על "toy example" הנקרא SPECTF.train הנמצא בקישור: https://archive.ics.uci.edu/ml/machine-learning-databases/spect/
- לחלק מהאלגורתמים קיים מימוש באינטרנט (github וכו'). מותר להשתמש במימוש קיים אך אז יש לציין זאת במפורש ויש לבדוק כל אלגוריתם כזה על 40 בסיסי נתונים (במקום 20 – ראו חלק ב')

חלק ב: מימוש האלגוריתם וניתוח ביצועים (50 נק')

1. ממשו את שני האלגוריתמים שתיארתם בחלק א'.
2. יש להשוות את האלגוריתם שבחרתם למימוש לאלגורתמים הבאים לצמצום מימדים: 1) mRMR 2) מדד f_classif עם False Discovery Rate=10% באמצעות הפונקציה RFE 3) SelectFdr עם מודל הערכה מסוג SVM. 4) ReliefF. רצוי להשתמש במימושים המתאימים ב- sklearn.feature_selection ו/או בחבילה scikit-feature (https://github.com/charliec443/scikit-feature) ו/או https://pypi.org/project/ReliefF

3. בסיסי נתונים להרצה: בטבלה למטה ניתן למצוא 12 תיקיות של בסיסי נתונים (לחצו על השם כדי לגשת לתיקיה). כל תיקיה כוללת מספר בסיסי נתונים. יש לבחור 5 בסיסי נתונים מ-4 תיקיות שונות, כלומר בסך הכל 20 בסיסי נתונים (כל הבעיות הן בעיות סיווג). רצוי לכלול בסיסי נתונים מגוונים לפי מספר גנים, מספר classes ומספר דגימות (שימו לב שבחירה מגוונת מהווה חלק מהציון). חלק מבסיסי הנתונים נמצאים ביותר ממאגר אחד ולכן אין לבחור בו פעמיים. שימו לב כי בחלק מהמאגרים ה-samples נמצאים בשורות ובמאגרים אחרים ה-samples נמצאים בעמודות. ה-Class לרוב נמצא בשורה/עמודה ראשונה או בשורה/עמודה אחרונה אך לעיתים הוא ימצא בקובץ נלווה (למשל בשם meta). פרט לנתוני ה-microarray וה-Class, לעיתים אפשר למצוא פרטי מידע נוספים על הדגימות כגון גיל או מין.

Corpus	Data Format	Reading data	Comments
scikit-feature datasets	MAT	import scipy.io mat = scipy.io.loadmat('file.mat'))	
ARFF	ARFF	from scipy.io import arff import pandas as pd data = arff.loadarff('yeast-train.arff') df = pd.DataFrame(data[0])	
Datamicroarray	RData OR CSV	import pyreadr result = pyreadr.read_r('mtcars_nms.rdata')	See more details on the data: https://github.com/ramhiser/datamicroarray
bioconductor	CSV		
Elvira	DBC	https://pypi.org/project/cantools/	https://leo.ugr.es/elvira/DBCRepository/
Microbiome Learning Repo (ML Repo)	Txt in OTU format	Look for otutable.txt and task.txt	
EfficientFS	Various format		
microbiomic data	CSV		
MicrobiomeHD	Txt in OTU format	https://amplicon-sequencing-pipeline.readthedocs.io/en/latest/running.html	https://github.com/cduvallet/microbiomeHD
Misc	Mostly CSV		הנתונים נאספו ממאמרים שונים
Amplicon_metagenome	Two excel files for each dataset: sample_data where you can		https://github.com/yangfenglong/mAML1.0

	find the class (Disease.Mesh.ID) And another OUT file which contains the gene expression values		
mAML_benchmark_datasets	Two excel files Suffix mf for label and No suffix for the gene expression data		https://github.com/yangfenglong/mAML1.0

4. לכל בסיס נתונים רצוי להריץ תחילה טיפול בערכים חסרים במשתנים **המסבירים** (למשל SimpleImputer), קידוד כל המשתנים הקטגוריילים למספריים (אם קיים), הסרה של משתנים עם שונות אפס (VarianceThreshold) ולבסוף נירמול (PowerTransformer). ערך חסר במשתנה **המטרה** (אם קיים) מתייחס לדגימות שנלקחו מאנשים בריאים (לצורך בקרה) ויש להתייחס אליהם כמו class לכל דבר.
5. אלגורתמים ללמידת סיווג: יש להשוות את השפעת אלגורתמי צמצום המימדים על 5 אלגורתמים ללמידת סיווג, כדלקמן: (NB, SVM, LogisticsRegression, RandomForest, k-nearest neighbors (K-NN
6. יש למדוד את מדדי הדיוק הבאים: ACC, MCC ו-PR-AUC, AUC. יצוין כי בבעיות Multi-class יש לחשב את ה- micro-average(ראו לדוגמא: (https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html)
7. זמן הריצה לביצוע ה- feature selection כפי שנמדד בהפעלת השיטה FSMethod (ראה למטה)
8. זמן הריצה הנדרש לאימון מודל הלמידה – כפי שנמדד בהפעלת שיטת fit
9. זמן הריצה הנדרש ל-Inference (כלומר testing) בממוצע לכל רשומה – כפי שנמדד בהפעלת predict_proba
10. יש לאמן/לבדוק את מודל הסיווג עבור מספר משתנה של top_k features כדלקמן: k= 1,2,3,4,5,10,15,20,25,30,50,100
11. יש לבחור את סוג ה-Cross-Validation לפי גודל בסיס הנתונים כדלקמן:

סוג Cross Validation	**מספר דגימות**
Leave-pair-out	50>
(LOOCV (leave-one-out	50-100
Folds CV10	100<
5FoldsCV	1000<

12. מבנה הפלט של הניסויים

Dataset Name	Number of samples	Original Number of features	Filtering Algorithm	Learning algorithm	Number of features selected (K)	CV Method	Fold	Measure Type	Measure Value	List of Selected Features Names (Long STRING)	Selected Features scores

Example	30	7000	mRMR	LogisticsRegression	4	Leave-pair-out	10	AUC	0.95	Gen7,Gen20,Gen143,Gen9	0.9,0.5,0.2,0.15
			ראו סעיף 2	ראו סעיף 5	ראה סעיף 10	ראו סעיף 11		ראו סעיף 6 עד 9			

חלק ג' (20 נק')

הציעו שיפור לאחד משני האלגוריתמים שבחרתם, תארו את השיפור ובדקו את ביצועיו בהתאם להוראות של שאלה 1. השתמשו ב-New : prefix_ כדי להבדיל בינו לבין האלגוריתם שממשתם בשאלה הראשונה. הוסיפו את התוצאות לטבלת הפלט שתוארה בסעיף א.

חלק ד' (10 נק')

1. לכל בסיס נתונים ציינו את הקונפיגורציה שהביאה לתוצאת ה-AUC הטובה ביותר (כלומר מה שיטת ה-Feature selection, כמות ה-Features ושיטת הסיווג שהיו הטובים ביותר)
2. בהתאם לקונפיגורציה "המנצחת" של כל בסיס נתונים, הרחיבו (augment) את בסיס הנתונים כדלקמן:
   a. הפעילו את ה-FS בקונפיגורציה "המנצחת"
   b. לאחר ה-FS הפעילו את KernelPCA מסוג linear וגם KernelPCA מסוג rbf על ה-X של קבוצת **האימון** באמצעות הפונקציה: fit_transform. יש לשרשר את הייצוג המתקבל משני ה-Kernel לעמודות שהתקבלו לאחר שלב a.
   c. בצעו את הטרנספורמציה שנקבעה בסעיף הקודם גם על קבוצת **הבדיקה** (ללא אימון מחדש) באמצעות הפונקציה transform
   d. הרחיבו את X **וה-y** של **האימון בלבד** באמצעות **אחת** מהשיטות הבאות לבחירתכם:
      i. GAN באמצעות חבילת: https://pypi.org/project/datomize/
      ii. SMOTE (או אחד משיפוריו כגון BorderlineSMOTE לבחירתכם) כפי שניתן למצא בחבילה imbalanced-learn
   e. הפעילו את אלוגריתם האימון שנבחר לפי הקונפגירוציה "המנצחת".
   f. דווחו את התוצאות בטבלה שתוארה בסעיף א בהתאם לפרוטוקול CV המתאים. השתמשו ב-Aug : prefix_ כדי להבדיל בין מודל זה למודלים הקודמים.
   g. רשמו את מסקנותכם.

חלק ה' (5 נק')

השתמשו במבחן Friedman כפי שהוצג בהרצאה כדי לקבוע האם ההפרשים במדד ה-AUC מובהקים סטטיסטית בין שיטות ה-Filtering השונות. אם התוצאות מובהקות סטטיסטית (השערת האפס נדחתה) המשיכו במבחן Post-Hoc כדי לבחון את ההפרשים בין האלגוריתמים. כתבו את מסקנותיכם.

חלק ו' (5 נק')

קראו את המאמר

Airola, Antti, et al. "An experimental comparison of cross-validation techniques for estimating the area under the ROC curve." *Computational Statistics & Data Analysis* 55.4 (2011): 1828-1844.

הסבירו מדוע פרוטוקול הבדיקה LOOCV (leave-one-out cross validation) עשוי להעריך בחוסר (underestimate). רמז: בדקו כיצד המסווג DummyClassifier פועל עבור בעיית סיווג בינארית מאוזנת במקרה של LOOCV. הסבירו כיצד הפרוטוקול Leave-pair-out מתמודד עם הבעיה.

<u>הוראות כלליות</u>

1. **שפת המימוש**

יתקבלו מימושים בשפות הבאות: Python (מומלץ) או R

2. **פוקנציות עיקריות.**
כל אלגוריתם לבחירת משתנים יוגדר על פי ה-API המקובל של scikit-learn בספריה sklearn.feature_selection באמצעות הגדרת פוקנציה במבנה הבא:

(....,YourAlgorithm (X, y

תוכלו להוסיף לפונקציה היפר-פרמטרים נוספים בהתאם לצורך.
הפונקציה תחזיר וקטור באורך של מספר התכונות ב-X ולכל תכונה את ה-Score המתאים לפי האלגוריתם.
ראו לדוגמא כיצד הגדירו את הפונקציה r_regression בקישור:
https://github.com/scikit-learn/scikit-learn/blob/baf828ca1/sklearn/feature_selection/_univariate_selection.py#L230

אם האלגוריתם אינו מחזיר Score או דירוג לכל תכונה, אזי יש להחזיר וקטור בינארי עם הערך 1 לכל תכונה שנבחרה ואפס אחרת.

את הפוקנציה שלכם תוכלו להפעיל באמצעות הפונקציה ה-transformer:
feature_selection.SelectKBest([YourAlgorithm, k])

3. את כל התהליך כולל ה-preprocessing, בחירת ה-features ואימון המודל, הגדירו כ-pipeline (ראו הסבר בקישור: https://scikit-learn.org/stable/modules/compose.html). כדי לחסוך בזמן ריצה, שימו לב כי ניתן להשתמש ב-Caching ב- pipeline וכן GridSearchCV לצורך הרצה של מספר ניסויים (את בחירת האלגוריתם הלמידה אפשר לעשות על ידי הגדרת Classifier Switcher כפי שמודגם בקישור:
https://stackoverflow.com/questions/50285973/pipeline-multiple-classifiers
4. אם זמן הריצה באלגוריתם שבחרתם גדול מידי, אפשר (אך לא מומלץ) לבצע "סינון מהיר" של גנים (למשל באמצעות מדד f_classif) כדי לאתר את ה-1000 הגנים החשובים ביותר. אם החלטתם לבצע סינון זה

בבסיס נתונים מסוים, יש להפעילו כ-preprocessing עבור כל השיטות שאתם בוחנים בבסיס נתונים זה. כמו כן ציינו בדו"ח עבור איזה בסיס נתונים השתמשתם בסינון זה.

5. את איסוף התוצאות מומלץ (לא חובה) לבצע באמצעות החבילה weights & biases. ראו לדוגמא: https://colab.research.google.com/github/wandb/examples/blob/master/colabs/scikit/Simple_Scikit_Integration.ipynb#scrollTo=PusIQpdPzUbP
6. אם הקוד שלכם רץ על גבי מעבדים של אינטל מומלץ להשתמש ב-patch הבא לצורך קיצור זמן הריצה באופן משמעותי: https://intel.github.io/scikit-learn-intelex/

7. **מה משפיע על הציון? (מהמרכיב החשוב ביותר לפחות חשוב, יחושב בנפרד לכל חלק בפרויקט בהתאם לרלוונטיות)**

- שלמות – כל מרכיבי הפרויקט קיימים.
- נכונות.
- בהירות - כתיבה מובנת של הדו"ח
- מורכבות הפרויקט. – האם בחרתם באלגוריתמים פשוטים או מורכבים?
- גיוון ב-dataset שנבחרו לבדיקה.
- הקף הניסויים שבוצעו.
- הסקת מסקנות מעניינות הנתמכות בתוצאות הניסויים.
- תיעוד הקוד
- מקוריות.
- יעילות המימוש.

8. **עבודה מקורית**

כל עבודה המוגשת על-ידכם (דו"ח, תרשים, נוסחה, שקף, קוד תוכנה וכו') אמורה להיות עבודתכם האישית אלא אם כן צוין אחרת ע"י הפניה מפורשת למקור המידע. מותר להעתיק קטעים מהמאמר, אך אז יש לציין זאת במפורש. <u>ניסיון להגיש עבודת אחרים כעבודתכם האישית (plagiarism) עלול להביא לציון אפס בקורס.</u> אם העבודה משמשת גם לקורס אחר או משתמשת בסיס לעבודת הפרויקט הסופי – יש לציין עובדה זו **<u>במפורש</u>** וכן להדגיש אלו תוספות נבעו מקורס הנוכחי.

9. **אופן ההגשה**

- קוד המקור. כמו גם הוראות מפורטות להרצה של הקוד (רצוי באמצעות מחברת מתאימה). פרטו בתיעוד הקוד את הייפר-פרמטרים של האלגוריתמים . העלו את הפרויקט שלכם ב- GitHub. הפרויקט חייב לכלול את כל פעולות pre-processing/post-processing על גבי הנתונים.
- דו"ח מסכם ב-Word או Latex הכולל תיאור ההסברים והניתוחים הנדרשים בחלקים האחרונים. בקובץ הזה תוכלו לראות מבנה מומלץ לדוח המסקנות.
- קובץ RAR או ZIP להגשה בתיבת ההגשה המתאימה במודל. הקובץ יכלול:
  - דו"ח על פי חלקי הפרויקט – יש לציין את חברי הקבוצה בעמוד השער
  - תוצאות מפורטות של הניסויים בפורמט אקסל xlsx במבנה הטבלה לעיל.
  - קבצי הנתונים שהשתמשתם בפרויקט **לאחר** פעולות העיבוד המקדימות. (ניתן לשמור בפורמטCSV)

- קישור לכתובת ה-GitHub המתאימה
- קוד מקור (בנוסף ל-gitHub) – קובץ זה ישמש לצרכי גיבוי
- קישור לסרטון קצר (עד 10 דק') שבו תציגו את עבודתכם. בפרט רצוי לתת דגש בסרטון על השיפורים שהצעתם בחלק ג' והמסקנות שהסקתם.