**LEBANESE AMERICAN UNIVESRITY**

**COMPUTER SCIENCE AND MATH DEPARTMENT, BYBLOS**

**CSC615 MACHINE LEARNING**

**ASSIGNMENT #1  
  
  
Ghadi El Hasbani, LAU I.D.: 201903122**

**Danielle Antoun Azar, PhD**

1. RESEARCH (10 points) Locate 5 conferences in Machine Learning and 5 journals. The core of the conferences and the journals should be Machine Learning (for example, you cannot choose a journal on Image Processing because it is a field where Machine Learning is applied). For each conference, list each of the following: the title, the dates, the submission deadline, the location where the conference will take place next, and a reference to the website. For each journal, list each of the following: the editor, the turnaround time, the impact factor, the quartile to which it belongs, and a link to the website of the journal.

Conferences:

1. International Conference on Machine Learning (ICML):

* Date: July 24 – 30, 2023
* Location: Seoul, South Korea
* Submission Deadline: Not announced yet
* Link: <https://icml.cc/>

1. European Conference on Machine Learning and Principles of Knowledge Discovery in Databases (ECML PKD):

* Date: September 19 – 23, 2022
* Location: Grenoble, France
* Submission Deadline: March 30, 2022
* Link: <https://2022.ecmlpkdd.org/>

1. International Conference on Machine Learning and Data Mining (MLDM):

* Date: July 15 – 19, 2023
* Location: New York, USA
* Submission Deadline: January 15, 2023
* Link: <http://www.mldm.de/>

1. International Conference on Frontiers of Artificial Intelligence and Machine Learning (FAIML):

* Date: April 28 – 30, 2023
* Location: Hangzhou, China
* Submission Deadline: November 8, 2022
* Link: <https://www.faiml.org/>

1. International Conference on Machine Learning and Applications (ICMLA):

* Date: December 12 – 15, 2022
* Location: Nassau, The Bahamas
* Submission Deadline: July 18, 2022
* Link: <https://www.icmla-conference.org/>

Journals:

1. Journal of Machine Learning Research (JMLR):

* Editor(s)-in-chief:
* Francis Bach, INRIA
* David Blei, Columbia University
* Turnaround Time: 4-6 months, 200 days (median)
* Impact Factor: 5.413
* Quartile: Q1
* Link: <https://www.jmlr.org/>

1. Machine Learning, Springer:

* Editor(s)-in-chief:
* Hendrik Blockeel, KU Leuven
* Turnaround Time: 47 days (median)
* Impact Factor: 4.541
* Quartile: Q1
* Link: <https://www.springer.com/journal/10994>

1. International Journal of Machine Learning and Cybernetics, Springer

* Editor(s)-in-chief:
* Xi-Zhao Wang
* Daniel S. Yeung
* Turnaround Time: 23 days (median)
* Impact Factor: 4.377
* Quartile: Q2
* Link: <http://www.ijmlc.org/>

1. Foundations and Trends in Machine Learning:

* Editor(s)-in-chief:
* Michael Jordan, University of California
* Turnaround Time: Not Found
* Impact Factor: 13.21
* Quartile: Q1
* Link: <https://www.nowpublishers.com/MAL>

1. Machine Learning: Science and Technology, IOPScience:

* Editor(s)-in-chief:
* Michael Jordan, University of California
* Turnaround Time: 46 days (median)
* Impact Factor: 6.013
* Quartile: Q2
* Link: <https://iopscience.iop.org/journal/2632-2153>

If quartiles or impact factors were not found on the journal website, scimagojr.com was used since LAU’s Journal Citation Reports (JCR) access was not working.

1. RESEARCH/APPLIED (10 points) Find three data set repositories (other than Kaggle and UCI repository) that have public data available. List them and give their URL. Pick one data set from one of them. Pick any one of your choice as long as it has the following characteristics:

* classification labels: binary
* total number of instances is greater than or equal to 8,000
* number of features (excluding the classification label) between 5 and 20.
* balanced data set (the ratio of one class around 60/100)

Describe the data set:

* what does it describe (e.g. patients suspected of having melanoma, etc.)?
* how many instances?
* what are the known attributes and what is unknown attribute? The type of each (real, string, integer, etc.)
* classification label distribution (e.g. 140 positive, 160 negative)
* three most recent publications that used these data sets (give a link to the publication and reference it). List the obtained results and the technique used.
* OpenNeuro: <https://openneuro.org/>
* GEO Datasets: <https://www.ncbi.nlm.nih.gov/gds>
* Cell Image Library: <http://www.cellimagelibrary.org/pages/datasets>
* GitHub: <https://github.com/>

Steinmetz Mouse Data (Steinmetz et al., 2019): <https://github.com/nsteinme/steinmetz-et-al-2019>  
  
The dataset describes neurological and behavioral responses of 10 mice undergoing 39 sessions of a Go/NoGo task for a total of 10050 trials (instances). The Go/NoGo task corresponds to a simple cognitive task where a mouse is positioned in front of three screens with access to a wheel and a feeding tube. The screens present a visual Gabor stimulus which appears to the left and/or the right screens. The mouse is expected to turn the wheel in the direction with greater contrast (a Go Trial) after a Go cue. When no stimulus appears, the mouse is expected to keep the wheel centered (a NoGo Trial). When the mouse responds as expected, a positive feedback in the form of food is given through the tube. Otherwise, a short 1s white noise burst is delivered for negative feedback. Since the trials are almost equally distributed between left, right, and NoGo, a binary label to indicate the Go/NoGo status of the trial would not yield a balanced dataset, but would have a ratio of around 2/3 as compared to 68.89% positive trials for the feedback type. Moreover, considering only left and right trials would cost 3305 trials. Primarily for these reasons the binary label considered is the Go/NoGo, which is positive for around 66.96% of the trials. The dataset is then downsampled by 2000 positive feedback trials to yield 8050 trials (instances) with around 58.75% positive feedback.

The neurological responses include spikes for 29,134 individual neurons, from which firing rate can be derived, localizable in a total of 42 brain regions as well as Low Frequency Potentials (LFPs) as recorded by neuropixels. On the other hand, the behavioral data includes face motion, wheel speed, pupil size, and pupil 2D coordinates with time as well as reaction time, lick detection, and feedback type for each trial. All time-series are binned at 10ms. Each trial starts 500ms (first 50 observations) before and ends 2000ms after stimulus onset. Each stimulus is shown for approximately the whole latter 2000ms of a trial with a minimum of 1000ms inter-trial interval. Since the neurological data corresponds to multiple brain regions that are inconsistent between mice, it would be difficult to fit a meaningful model on all trials. For this reason, behavioral features, characterized as the similarity measure (numerical and continuous real values) with respect to the top 4 shapelets of the shapelet transform, as described by Lines et al. (2012, August), of each the face motion, wheel speed, pupil size, and pupil 2D coordinate time-series, are used, yielding a total of 4\*5 = 20 features. The transform was performed after data splitting into 20% of mice for testing, then scaled, and 80% of mice for training. The splitting was done over mice so that the test set is completely independent since the model will not be fit over neural data from these mice. Although this does not guarantee that the test set will contain 20% of instances, the testing metrics will be better indicators of generalizability. The transformed training data is then scaled for each fold separately. The scaling is done because the similarity metric, mutual information, outputted by the shapelet transform only has a lower bound at 0 but no upper bound. For the sake of running time, all time-series were also downsampled with a low-pass mean-pooling filter of non-overlapping windows of size 10. This reduced the number of time-points of each time-series from 250 to 25.

The Steinmetz dataset was collected and first used by Steinmetz et al. (2019). Wilcoxon sign-rank or rank-sum tests were used to assess significance of neuronal modulation between different conditions such as baseline vs trials, baseline vs post-stimulus onset, baseline vs post-movement, baseline vs post-reward on NoGos, or left vs right trials. Significance of neuronal visual receptive fields, determined by sparse noise mapping, were also assessed using a Wilcoxon rank-sum test between rates of a neuron at peak location and locations further than a threshold. To identify choice-selective neurons, Kernel (choice, action, and vision kernels) reduced-rank (rank = 18) regression with elastic-net regularization (alpha = 0.5, lambda = 0.5) was used. The firing rate at a time t (response) is represented as a linear sum of temporal filters over binned time-points, time of movement and time of appearance of a contrast stimulus, for each neuron yielding N models. K-Fold cross-validation (K = 5) was used to estimate model performance using percent variance explained and proportion of overfit explained variance as metrics. The reduced-rank regression outperformed both regular regression and a model with raised cosine basis functions. Neuron selectivity was determined by fitting the reduced-rank regression on the firing rate using all but one kernel to test followed by fitting a reduced-rank regression on the residuals of the former model using the tested kernel only. The cross-validated variance explained by the latter model would considered as explained only by the tested kernel, and a neuron is selective for the tested kernel if the metric is greater than 2%. The focality index was then derived as the proportion of neurons selective for a particular task (kernel) in a specific area. The selectivity of neurons using this model was consistent with biological relevance of the neuron and its location. Moreover, a GLM with binomial link function was fit using behavioral predictors including inter-trial interval, previous trial feedback type, as well as z-scored pre-stimulus pupil area and motion to predict whether a mouse will respond to a trial. The squared terms of the last two predictors were also included and trials were matched between contrast conditions. To test whether engagement index predictor would improve performance, a deviance test was calculated between models with and without this predictor. The engagement index significantly contributes to the model, an effect unexplained by the other predictors.

Stringer et al. (2021) split the data into 75% training and 25% testing by taking every 4th trial to be in the testing set. Tuning curves were then computed from stimulus angle (radians) by fitting the training trials to n = 10 sine and cosine basis functions. A linear regression was then fit over those functions to predict stimulus responses defined as the summed activity of around 650ms timeframe after stimulus onset. Generative models for the neural data were built by modeling the mean of each neuron as a function of stimulus angle using the tuning curves. The mean predictions were then subtracted and residuals were squared and used with the same basis functions to model the standard deviation of each neuron. Using those definitions for the mean and standard deviation, an independent decoder, naïve Bayes classifier, was then built to decode a discrete set of stimulus orientation (n = 48) log-probabilities, summed across neurons and upsampled using kriging interpolation, that a novel neural pattern is observed for a putative stimulus angle using a normal distribution. The peak of the upsampled curve corresponds to the predicted orientation, and leave-one-out cross-validation was performed with bootstrap (n = 100) estimates of error bars. Decoding erros for the model proved to be correlated, violating assumption of independence. Moreover, neurometric fitting was performed by fitting a linear decoder for the high density stimulus set to predict angles in a small range (4°). The linear decoder was built by regressing neural activity onto von Mises tuning curves (n = 48, σ = 0.1) with peaks evenly spaced along 360°. The transformation is fit using ridge regularization constant of 1, and the responses were upsampled. The model outperforms the independent decoder and two other common approaches. Performance was robust to spontaneous activity patterns and reductions in dimensionality ≥ 128 but could be improved with more trials. For the low density stimulus set (range of ± 30°), a difference of von Mises basis functions of the stimulus angle and threshold angle was first predicted, and the sign of the prediction was used as the predicted class label. Discrimination performance was then calculated as a probability function, fit using centered sigmoids, representing the fraction of correct predictions for a given angle bin to give neurometric curves averaged over multiple evenly spaced values of threshold angle and symmetrized. Discrimination thresholds showed a wide discrepancy between neural and behavioral responses, with neural thresholds reaching as low as 0.34° ± 0.04°.

In Khajehnejad et al. (2022), a feed-forward neural network architecture of 3 components is used to encode the stimulus at a given trial into a neural then behavioral response. The first component maps the stimulus at a trial into a vector through 2 fully connected layers with 20 units (soft-plus activation) followed by a layer with 50 units (linear activation). The elapsed time since stimulus is passed through a linear layer of 50 units whose output is then added to the former linear layer and passed through a tanh activation unit. The second component maps the tanh activation unit output into neural activity cumulative intensity functions through 3 20 unit dense layer with tanh activation and a 42 unit (as many as brain regions) dense layer with soft-plus activation, in that order. The third component finally maps the neural activity to behavioral responses/actions as cumulative intensity functions. Each response has a subcomponent composed of 1 dense layer with 10 units that receives the neural activity input and an output layer with 1 unit. Middle layers use tanh activation functions, output layers use soft-plus, and weights are constrained to be positive for both second and third components to ensure they behave monotonically. All weights from stimulus to neural activity are trained with Adam optimizer using the loglikelihood computed for spike trains (10 steps) whereas the remaining weights were trained, also using Adam, with the loglikelihood computed on the behavioral data (420 steps) until no significant improvement is achieved. The model’s second component produced biologically-relevant neural activity that is region- and time-specific irrespective of stimulus and motion direction. The third component’s behavioral response timing also correlates with biologically-relevant region-specific activity irrespective of response type. In general, the model outperforms baseline models such as GLM, NHPoisson, Universal Count Model, and P-GPL.

1. APPLIED (30 points) The most recent descendant of ID3 is C5. Use it to construct decision trees using the data set you selected in Question II. Let’s call it data set A. Take out 20% of the instances and form set C. Remove the instances from A. Set C is your testing set. Take set A (the smaller/modified one) and perform 10 fold-cross validation on it. This gives you 10 pairs of folds (ai, bi) where ai is a training fold and bi is a validation fold. Run C5 with the following window sizes to generate 10 trees in each case: *w*=20, *w*=40, *w*=60, *w*=80
   1. For each of the values of *w* above do the following:
2. for each of the 30 trees, compute the metrics listed below on each of the training and validation folds ai and bi and on C.
3. Tabulate the results into three columns: training, validation and testing. Compute the average and standard deviation of the training, the validation and testing metrics. These are the training/testing/validation metrics of C5 given a particular window size. Note any difference you see between the training, the validation and the testing metrics.

Metrics: Precision, recall, specification, sensitivity, confusion matrix, AUC, F1-measure

* 1. Plot the metrics computed in the previous step, explain how they change with window size.
  2. Convert the trees to rule sets (using C5 built-in command). Is there any attribute that never appears in the rule sets for any particular window size? Is there any relationship between the window size and the size of the rule sets?
  3. Remove all cases with missing values from ai, bi and C. Re-construct the trees choosing the value of *w* that gave you the best validation accuracy. Repeat from Step 1. Any change in numbers? Explain why or why not.

1. Since the dataset did not contain any missing values, they were simulated using the Missing Completely at Random (MCAR) procedure with a probability of 0.1. All metrics are averaged over 300 trees given a particular value of nTrials and set of instances.

Testing metrics are almost consistently lower than training and validation metrics. This is plausible considering that the testing set contains behavioral data from mice not included in the model’s training data, whereas k-Fold CV splits are based on shuffled trials. This means that training and validation sets rarely contain data from distinct sets of mice. This shows that although the model performs fairly well on the same mice it was trained on, its performance on new mice is weak. This is especially true when it comes to the proportion of NoGo predictions that are correct, specificity, as the model seems to have a high false positive rate. It might be that the model fails to distinguish between behavioral responses that are indicative of a Go trial decision, and those that occur passively during a NoGo trial. Nevertheless, the relatively high testing precision indicates the model performs well with regards to true positives.

1. The results for the dataset with missing values show that the precision slightly increases for the testing set and decreases for training and validation sets with increasing number of boosting trials (nTrials). Nevertheless, although the precision becomes comparable at nTrials = 80, the increase in the testing set precision is minimal and accompanied by a major increase in standard deviation (SD). The proportion of the model’s positive (Go) predictions that are correct is not drastically improved by increasing nTrials.

Training and validation specificities slightly increase with nTrials whereas the testing specificity drops at nTrials = 60. The SD also increases drastically with nTrials. This suggests that the proportion of the negative (NoGo) trials correctly detected by the model is slightly overfit with increasing nTrials.

Recall for all sets decrease almost equally accompanied by an increase in SD with increasing nTrials. This suggests the proportion of positive (Go) trials correctly detected by the model is not improved but made worse with increasing nTrials. AUC and its SD remain almost constant across values of nTrials with training having the highest AUC, followed by validation then testing AUCs. This metric suggests that the predictive performance of the model is not improved or made worse with increasing nTrials.

The F1 score decreases for all sets accompanied by increasing SD with increasing nTrials until nTrials = 60 after which the testing F1 score increases above training and validation F1 to a value slightly lower than that of the testing F1 score at nTrials = 20. The F1 score suggests that the detection of Go trials by the model can possibly be improved by increasing nTrials above 80. Nevertheless, this increase is possibly due to the increase in precision which seems to overcome the simultaneous decrease in recall at nTrials = 80 for the testing set. This is also evidenced by the decrease in the absolute value of the slope of the testing recall curve after nTrials = 40. This suggests that this effect might not be completely relevant especially given that most metric SDs suggest a greater performance variability in values of nTrials > 20.

1. Every attribute appears at least one with any given number of trials. There is no relationship between number of trials and the number of occurrences of any given attribute. There is also no relationship between ruleset size and number of trials.
2. For the original dataset without missing values, the best validation accuracy yields an value of nTrials = 20. All reported metrics are averaged over 300 trees.

Testing metrics, especially specificity give much lower results as compared to training and validation which is consistent with the MCAR dataset. This means that the model’s generalizability is generally noticeably low, but more so when it comes to the proportion of NoGo predictions that are correct as it has a high false positive rate, specificity (0.57 ±0.02 for testing). It might be that the model fails to distinguish between behavioral responses that are indicative of a Go trial decision, and those that occur passively during a NoGo trial. Nevertheless, the relatively high testing precision (0.68 ±0.06) indicates the model performs well with regards to true positives. This is also evident for training and validation sets. On the other hand, recall is relatively lower than precision for training and validation sets but is comparable to precision in the testing set (0.68 ±0.01). This could mean that false negative rate is higher than false positive rate in training and validation but are comparable in testing. This means that training and validating on the same mice gives the advantage of improved false negative rate as compared to false positive rate whereas new mice might have very different behavioral indicators of positive Go and negative NoGo trials which leads to the model underperforming in detecting both indicators. This also means that the weak model performance in differentiating between behavioral responses that are passive and active is an issue whether between mice or within the same mouse.

The values also barely differ from those of results using nTrials = 20 with the MCAR dataset. This shows that the mechanism that C5.0 uses to deal with missing values is adequate.

Each attribute appears at least once in any of the 300 trees for the best nTrials value. Size of rulesets also does not differ and has the same median value of 7 as with the MCAR dataset. Nevertheless, the outliers are no longer present because the median is now more centered in the interquartile range making the distribution more normal.

1. (10 points) We have seen in class C4.5 which is the direct ancestor of C5. Explain the major points that differentiate the two programs.

The rulesets methods of C4.5 consume a lot of memory. C5.0 was developed as a more memory and time efficient tree-fitting algorithm that is highly optimized and produces smaller trees than C4.5 (Pandya & Pandya, 2015). A boosting method that is partly robust to noise is also incorporated in C5.0 for improved predictive performance and generalizability as evidenced by lower error rates on test cases (Pandya & Pandya, 2015). The boosting method places attention on misclassified instances through raising their weights instead of the windowing approach of C4.5 (Quinlan, 1996, August). Multiple new parameters such as variable misclassification costs and case weight attributes are also available in C5.0 as well new data types such as timestamps and ordered discrete attributes. C5.0 also allows values to be indicated as Not Applicable. C5.0 also allows defining attributes as a function of others. To combat high dimensionality, C5.0 also uses winnowing to discard marginally relevant attributes through feature selection prior to tree fitting. Finally C5.0 is easier to use by joining tree and ruleset functionalities into one as well as support for sampling and cross validation.

<https://www.rulequest.com/see5-comparison.html>

**References**

Barrett, T., Troup, D. B., Wilhite, S. E., Ledoux, P., Rudnev, D., Evangelista, C., ... & Edgar, R. (2007). NCBI GEO: mining tens of millions of expression profiles—database and tools update. *Nucleic acids research*, *35*(suppl\_1), D760-D765.

Gorgolewski, K., Esteban, O., Schaefer, G., Wandell, B., & Poldrack, R. (2017). OpenNeuro—a free online platform for sharing and analysis of neuroimaging data. *Organization for human brain mapping. Vancouver, Canada*, *1677*(2).

Khajehnejad, M., Habibollahi, F., Nock, R., Arabzadeh, E., Dayan, P., & Dezfouli, A. (2022). Neural Network Poisson Models for Behavioural and Neural Spike Train Data. *bioRxiv*, 2020-07.

Lines, J., Davis, L. M., Hills, J., & Bagnall, A. (2012, August). A shapelet transform for time series classification. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 289-297).

Myint, K. L., & Tin, H. H. K. (2021). Analyzing the Comparison of C4. 5, CART and C5. 0 Algorithms on Heart Disease Dataset using Decision Tree Method. *ICIDSSD 2020*, 174.

Pandya, R., & Pandya, J. (2015). C5. 0 algorithm to improved decision tree with feature selection and reduced error pruning. *International Journal of Computer Applications*, *117*(16), 18-21.

Quinlan, J. R. (1993). Program for machine learning. *C4. 5*.

Quinlan, J. R. (1996, August). Bagging, boosting, and C4. 5. In *Aaai/Iaai, vol. 1* (pp. 725-730).

Steinmetz, N. A., Zatka-Haas, P., Carandini, M., & Harris, K. D. (2019). Distributed coding of choice, action and engagement across the mouse brain. *Nature*, *576*(7786), 266-273.

Stringer, C., Michaelos, M., Tsyboulski, D., Lindo, S. E., & Pachitariu, M. (2021). High-precision coding in visual cortex. *Cell*, *184*(10), 2767-2778.